Department of Computer Science
Series of Publications A
Report A-2017-5

# Natural language processing system for business intelligence

Mian Du

*To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium A111, Exactum building, on November 29th, 2017, at 12 o'clock noon.*

University of Helsinki
Finland

**Supervisors**

Professor Sasu Tarkoma, University of Helsinki, Finland
University Researcher Roman Yangarber, University of Helsinki, Finland

**Pre-examiners**

Professor Jan Snajder, University of Zagreb, Croatia
Professor Wei Xiong Rao, Tongji University, China

**Opponent**

Senior Lecturer Mark Stevenson, University of Sheffield, UK

**Custos**

Professor Sasu Tarkoma, University of Helsinki, Finland

**Contact information**

Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: info@cs.helsinki.fi
URL: http://cs.helsinki.fi/
Telephone: +358 2941 911, telefax: +358 9 876 4314


ISSN 1238-8645
ISBN 978-951-51-3900-9 (paperback)
ISBN 978-951-51-3901-6 (PDF)
Computing Reviews (1998) Classification: I.2.7, H.3.3, H.2.8
Helsinki 2017
Unigrafia

**Natural language processing system for business intelligence**

Mian Du

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
du@cs.helsinki.fi
http://cs.helsinki.fi/Mian.Du/

PhD Thesis, Series of Publications A, Report A-2017-5
Helsinki, November 2017, 78+72 pages
ISSN 1238-8645
ISBN 978-951-51-3900-9 (paperback)
ISBN 978-951-51-3901-6 (PDF)

**Abstract**

The ongoing information explosion has a particular impact on business areas, involving corporate strategy and business decision-making. Business intelligence tools aim to help users to understand market trends, which is critical for their day-to-day operations. For example, it is a typical business intelligence task to effectively obtain accurate and relevant information about the competitor's activity in the same industry sector. This thesis presents research on a natural language processing system, which aims to address the problem of information overload in the business domain. It uses document filtering, information extraction, and supervised and semi-supervised learning. Input to the system includes news documents from on-line news websites and company press pages.

We first demonstrate that a combination of NLP techniques and frequent sequential pattern mining can be used for finding patterns from unstructured natural-language text, i.e., news articles. The patterns relate to a specific domain of news. Evaluation results show that scenario-based summarization can filter out irrelevant documents and also extract important sentences from relevant documents as summaries for pre-defined scenarios in a specific domain. For document-level filtering, this method achieves very high precision, while keeping quite high recall in our study.

Next, we present experiments with supervised learning for labelling business-news documents with multiple industry sectors. The main contribution

is that combining a named-entity-based rote classifier with the balanced classifiers yields better results than either classifier alone. This method also improves on the best score previously reported, while using the same amount of training data for the rote classifier, and considerably less for the statistical classifiers.

We then explore the interplay between company news, social media visibility, and stock prices. Information extracted from on-line news by means the of deep linguistic analysis is used to construct queries to various social media platforms. The main results presented in the thesis demonstrate the interesting correlations between the mentions of a company in the news and the views of its page in Wikipedia.

Based on the above research topics, the thesis also presents the design and architecture of a complete decision-support system. The system is an example of using the above research results to extract, analyze and organize information from plain-text news.

**Computing Reviews (1998) Categories and Subject Descriptors:**
I.2.7 [Artificial Intelligence]: Natural Language Processing - Text analysis
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval
H.2.8 [Database Applications]: Data mining

**General Terms:**
Natural Language Processing, Information Extraction, Machine Learning, Decision Support

**Additional Key Words and Phrases:**
Information Explosion, Business Intelligence, Information Retrieval, Document Filtering, Supervised Learning, Sequential Patterns, Information System

# Acknowledgements

How time flies! Looking back to nine years of Master's and doctoral studies in Finland, there has been joy of success, there has been frustration of failure, there are many people and things worth remembering. First of all, I would like to express my sincere gratitude to my supervisor Roman Yangarber for accepting me as an research assistant in PULS project, for the continuous support of my Master's and doctoral studies and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not go this far without him.

I am really grateful to my supervisor Professor Sasu Tarkoma. This research and dissertation was done with kind attention and guidance from him. His serious scientific attitude, rigorous scholarship, working style of excellence, deeply affected and inspired me.

I would very much like to thank my pre-examiners Professor Jan Snajder and Professor Wei Xiong Rao for providing me precious review statements to help me improve the thesis in all aspects. Their insightful reviews have lit the way and have strengthened my desire to further conduct this research.

I am very grateful to my colleagues. I am really enjoying the team work with Roman Yangarber, Lidia Pivovarova, Matthew Pierce, Peter von Etter, Ossi Karkulahti, Jussi Kangasharju, Arto Vihavainen, Silja Huttunen, Mikhail Novikov, Natalia Tarbeeva and many other talented researchers. Their professional attitude toward work has inspired and supported me during my Master's and doctoral studies. We have created a great information extraction system together and many scientific results.

Very special thanks to Doctoral Programme in Computer Science (DoCS) and the Helsinki Doctoral Programme in Computer Science (Hecse) for their financial support wit travel grants. I had the opportunity to attend important conferences, summer schools and other events to broaden my network, improve my research through communication and study, and visit many interesting places. Thanks to Pirjo Moen and Petri Myllymäki for their arrangement of this support.

Experiments in this research are mostly completed using the cluster machines (Ukko) of the Computer Science Department at the University of Helsinki. During the experiments, we got great support and help from Jani Jaakkola, Pekka Niklander and all other IT support team members. Many thanks to all of you!

At the imminent completion of my study, I would like to express my great thanks to my parents for their endless love, support, guidance and encouragement for my study and life over the years. I am so lucky to be your son! I would like to thank all my relatives and friends for their great support during my study. Without them, I can not imagine how I would undertake this challenging career.

Last but not least, there are no words which can express my gratitude to He Zheng who agreed to be my wife, brought us a son, took care of us, enjoyed the challenging life with me. You are the best!

Helsinki, November 2017
Mian Du

# Contents

# List of Publications and Author's Contributions

The thesis contains six original publications, referred to as paper I-VI and reprinted at the end of the thesis. Other original publications to support the thesis are also referenced in the content of the thesis. Each reprinted paper describes the research work for part of the proposed system. My contribution to these papers is described below.

**Paper I (PI)**:

Mian Du and Roman Yangarber (2015), *Acquisition of domain-specific patterns for single document summarization and information extraction*, Proceedings of The Second International Conference on Artificial Intelligence and Pattern Recognition, Shenzhen, China.

M. Du designed, implemented and conducted the experiments. M. Du wrote the paper.

**Paper II (PII)**:

Mian Du, Matthew Pierce, Lidia Pivovarova and Roman Yangarber (2014), *Supervised Classification Using Balanced Training*, Proceedings of International Conference on Statistical Language and Speech Processing (SLSP 2014), Grenoble, France.

M. Du and R. Yangarber designed the classifier. M. Du implemented the classifier and M. Pierce improved the classifier and conducted the experiments. All authors supported the experiments. All authors wrote parts of the paper.

**Paper III (PIII)**:

Mian Du, Matthew Pierce, Lidia Pivovarova and Roman Yangarber (2015), *Improving Supervised Classification Using Information Extraction*, Proceedings of 20th International Conference on Applications of Natural

Language to Information Systems, NLDB 2015, Passau, Germany.

All authors designed the new algorithms and features for the classifier. All authors supported the experiments. All authors wrote parts of the paper.

**Paper IV (PIV)**:

Mian Du, Jussi Kangasharju, Ossi Karkulahti, Lidia Pivovarova and Roman Yangarber (2013), *Combined analysis of news and Twitter messages*, Proceedings of the Joint Workshop on NLP&LOD and SWAIE, RANLP 2013 Workshop on Semantic Web and Information Extraction, Hissar, Bulgaria.

All authors designed the experiments. M. Du implemented part of the experiments. M. Du wrote part of the paper.

**Paper V (PV)**:

Ossi Karkulahti, Lidia Pivovarova, Mian Du, Jussi Kangasharju and Roman Yangarber (2016), *Tracking interactions across business news, social media, and stock fluctuations*, Proceedings of 38th European Conference on IR Research, ECIR 2016 Padua, Italy.

All authors designed the experiments. M. Du implemented part of the experiments. all authors reviewed the paper.

**Paper VI (PVI)**:

Mian Du, Lidia Pivovarova and Roman Yangarber (2016), *PULS: natural language processing for business intelligence*, Proceedings of Workshop on Human Language Technology and Intelligent Applications, 25th International Joint Conference on Artificial Intelligence (IJCAI-2016), New York, USA.

All authors designed the system. M. Du was responsible for building the architecture of the system and implemented parts of the system and the decision-support system. M. Du wrote a significant part of the paper.

# Chapter 1

# Introduction

The ability of users to leverage information and convert it into actionable knowledge, getting the right information to the right people at the right time via the right channel is important for them to make strategic planning and informed decision making. This ability is supported by business intelligence (BI) tools. However, only preliminary parts of information collection for BI (e.g., document filtering, clustering and classifying) are done by machines according to an overview of language analysis applications in the business domain in this paper [1].

According to a survey conducted by Domo Technologies [2] in 2012, the consumerization of business intelligence is growing; the demands for the BI data to be more timely, more easily accessible, more easily visualized, as well as guaranteeing data integrity are considered to be the greatest challenges for BI professionals. Considering the rapidly growing number of raw data placed on the Internet, manual work for converting raw data to BI data is not a viable solution. In order to address the *information overload* issue and provide useful decision support in the business domain automatically, this research explores various methods to process raw on-line plain text into meaningful information for business users.

Document filtering (DF) is a technology often used in the beginning of text processing to retrieve relevant documents from a large dataset. This research proposes a method, which combines NLP techniques and frequent sequential pattern mining to filter irrelevant documents from a massive number of news documents on-line in real time. For document-level filtering in the business domain, this method achieves very high precision, while keeping high recall in our study.

Information Extraction (IE) is the main technology used to transform unstructured natural language text into pre-specified structured information [3]. The structured information can be stored into the database for

later query. Different methods of IE are described in Section 5.3.2. This research uses a pattern-based extraction method[1] to extract structured business information. The pre-specified structured information extracted by our IE patterns includes business activities such as corporate acquisitions, new product launches, investments, management-post appointments, etc. Each activity contains a number of attributes involved in the activity, such as companies, persons, location, business value, etc.

Machine learning (ML) is widely used to extract information from the unstructured data. This research experiments with supervised learning for labelling news documents with multiple industry sectors. We propose a method, which combines a named-entity-based rote classifier with the balanced supervised learning classifiers. The combination yields better results than either classifier alone. This method also improves on the best score previously reported.

Once we have extracted business information from news as stated above, we then explore the interplay between company news, social media visibility, and stock prices. Companies in business activities are used to construct queries to various social media platforms. We identified that the mentions of a company in the news and the views of its page in Wikipedia has interesting correlations.

The main research question is to study "How to address information overload and provide decision support in the business domain?" In order to answer this question, this thesis focuses on the following sub-questions:

- RQ1: How to filter irrelevant documents from many sources of continuously streaming news?
- RQ2: How to extract information not explicitly present in text from plain-text news?
- RQ3: How to link external information, e.g., social media visibility and stock prices with news?
- RQ4: How to construct the business decision-support system (DSS) using information from unstructured text data?

By integrating information extracted and processed by DF, IE and ML, we propose a combined approach to develop PULS Business Decision-

[1] An extraction pattern contains an indication for specific variable tokens and their surrounding context. While the surrounding context is fixed, the tokens are variable. An IE system usually has a large number of such extraction patterns to match required facts [4, 5, 6, 7, 8, 9, 10, 11].

Support System (P-BDSS)[2], as an example design and implementation that helps to solve these questions.

## 1.1 Methodology

Table 1.1 shows an overview of the research methodology used in this thesis.

| *Research Questions* | *Methodology* | *Publications* |
|---|---|---|
| RQ1: How to filter irrelevant documents from many sources of continuously streaming news? | Data collection, statistical data analysis, experiments and evaluation | PI |
| RQ2:How to extract the industry sectors of the business activities from plain-text news? | Data collection, Statistical data analysis, experiments and evaluation | PII & PIII |
| RQ3: How to link external information (e.g., social media visibility and stock prices) with news? | Statistical data analysis and case study | PIV & PV |
| RQ4: How to construct the business DSS using information from unstructured text data? | Requirements elicitation, evolutionary delivery model, modelling and evaluation | PVI |

Table 1.1: The methodology of this thesis.

PI collects relevant plain-text data and pre-processes the data into structured sequential data. We use news articles collected from business RSS as described in section 2.1.2. Statistical data analysis is used to understand the data and mine frequent sequential patterns. With a small amount of effort required for manual selection, these patterns can be used for domain-specific scenario-based document summarization and information extraction. Evaluation is performed to show that scenario-based document summarization can both filter irrelevant documents and create summaries for relevant documents within the specified domain. These patterns can also be converted into extraction patterns for IE.

PII and PIII collect information about the distribution of class labels over named entities found in text. Reuters corpus (RCV1) is used for our

[2]Decision-support system assists users in making decisions by utilizing data, models, knowledge and human-computer interactions provided by the system [12].

experiments. Statistical data analysis is used to understand the data. We then combine a knowledge-based rote classifier with statistical classifiers to obtain better performance than either classification method alone. The combined classifier achieves a significant improvement in macro-averaged F-measure [13] compared to the state of the art, while maintaining comparable micro-average.

PIV and PV present experiments and use case studies to demonstrate interesting correlations between news and social media contents. We focus on numerical measurement and analysis of the content. Nearly 4 million tweets collected using twiter api as described in table4.1 are used. We present three types of results. In the first experiment, we present the most frequently tweeted company or industry news by counting the number of Twitter posts, which contains a company name extracted from news. In the second experiments, we chose three companies—Alstom, Malaysia Airlines, and General Motors to perform visual analysis of correspondence between Wikipedia views, news hits and stock prices. In the third experiment, we choose eleven big companies from different industry sectors, namely Alibaba, Alstom, Burger King, General Motors, IBM, Malaysia Airlines, Medtronic, Mt. Gox, Netflix, The Home Depot, and Xiaomi. For each of these companies we collect two time series: daily news mentions and Wikipedia views. Then we calculate the cross-correlation between all possible pairs in these dataset to identify time-series correlations between news hits and Wikipedia views.

In PVI, requirements elicitation is used in a variety of sources including: project plans, literature sources, users' specific requirements, requirements from researchers' point of view. Since the requirements are not initially stable and new requirements always come up with research ideas and through continual interaction with end-users and project partners in the industry, it is best to use a model, which builds on successive prototypes to build P-BDSS. One example of such a model is called *Incremental Development Practices* (IDP) [14]. *IDP* refers to the development practices that allow a program to be developed and delivered in stages. IDP breaks the project into a series of small sub-projects, which are much easier to complete than a single monolithic project. Evolutionary Delivery Model, which is one of the life-cycle models that support incremental development has been chosen. Figure 1.1 illustrates how the system works [14]. By using the UML, which is one of the modelling languages, all functional requirements identified in the requirement stage have been transferred into actual functional designs. Finally, several comprehensive evaluation approaches are adopted to evaluate and test the new features from both functional and non-functional

point of view.

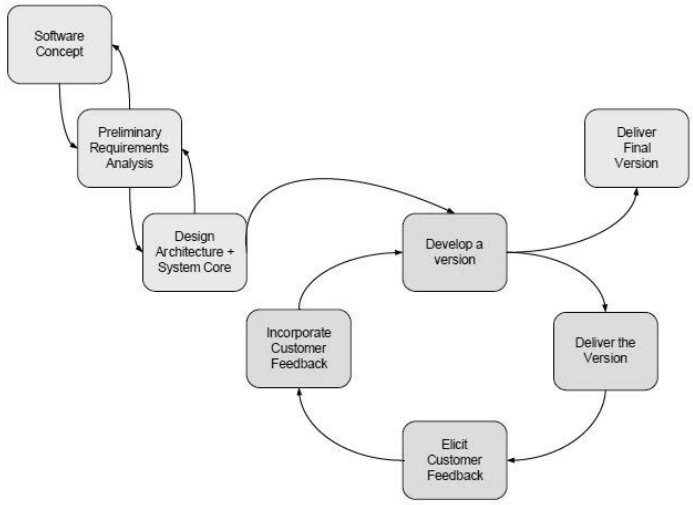



Figure 1.1: Evolutionary Delivery Model [14].

## 1.2 Thesis Contributions

This thesis contains six scientific articles as presented in Table 1.1. This section summarizes their contributions.

PI presents experiments on a pattern-based classifier to filter non-business news articles from a live business news corpus, in order to improve the quality of the corpus. This business news corpus is collected automatically by a document collection module as described in section 2.1. In an example case, we collected 100 random documents from this business news corpus, we have found that 30% of them are non-business documents, which contain no business activity. The non-business documents ratio goes up to 67% when we check 100 random documents collected from company websites. These non-business documents dramatically decrease the precision of the extraction result of IE. We therefore need to filter out non-business documents from a massive number of documents collected to improve the precision. PI demonstrates that a combination of NLP techniques and frequent sequential pattern mining can be used for finding patterns in a specific domain from unstructured natural-language text, i.e., news articles.

With a minimum manual selection effort, we use these patterns to generate domain-specific scenario-based document summaries. We have applied the method in two domains. The evaluation results show that scenario-based summarization can serve to filter out irrelevant documents and also extract important sentences from relevant documents as summaries for pre-defined scenarios in a specific domain. For document-level filtering, this method achieves very high precision while keeping quite high recall in both domains in our study. This demonstrates that this method may solve the problems for scenario-based document filtering in a specific domain.

PII and PIII present experiments with supervised learning to mark business news documents with multiple industry sectors. In many studies on the supervised classification, the traditional assumption is that not only the labels for test data have the same distribution of ones for training data, but also the classifier will be applied to the future data with the same distribution. However, this is not always the case: the label distribution may change over time, even in the same news stream. In addition, a single set of classifiers may need to mark data from multiple sources of news with different label distribution. We are interested in exploring the real world settings in which the distribution of labels may change over time. Therefore, one of our goals is to construct a classifier that does not favor a particular distribution in a given training set. Instead of using all available documents from the training set, we experiment with smaller subsets to balance the data. We use a balanced procedure that is suitable for multi-label settings. By using test sets with different label distributions, we demonstrate that the classifier trained on balanced data achieved better performance than the one trained on original distribution of the label in the corpus. The main contribution of these papers demonstrate that the combination of baseline rote classifiers on named entities and balanced supervised classifiers produces better results than any classifier alone. This method improves the best scores previously reported while using the same number of training data for rote classifiers, and fewer data for supervised classifiers. The experiment also shows combing company descriptors features from the knowledge base can not improve performance.

The complex relationships among traditional news, social media and stock price is an active research subject. Recent studies in the area have shown that it is possible to find some correlations between news and stock prices, when the news is properly categorized [15, 16]. We believe that the analysis of information can be of particular interest to experts in various fields: business analysts, Web scientists, data reporters and so on. PIV and PV present studies of the interplay between company news, so-

cial media visibility, and stock prices. Information extracted from on-line news by means of deep linguistic analysis is used to construct queries to various social media platforms. The main contribution of these papers is that we combine NLP with social media analysis, and discover interesting correlations between news and social media. The results presented in papers demonstrate the utility of collecting and comparing data from a variety of sources. In the first study, we have demonstrated that the impact that new-product events have on Twitter depends more on the industry sector than on a particular company. In the second experiment we chose three companies—Alstom, Malaysia Airlines, and General Motors to perform visual analysis of correspondence between Wikipedia views, news hits and stock prices. In the third experiment, we choose eleven large companies from different industry sectors, namely Alibaba, Alstom, Burger King, General Motors, IBM, Malaysia Airlines, Medtronic, Mt. Gox, Netflix, The Home Depot, and Xiaomi. For each of these companies we collect two time series: daily news mentions and Wikipedia views. Then we calculate the cross-correlation between all possible pairs in these datasets to identify time-series correlations between news hits and Wikipedia views.

PVI presents the design and architecture of a DSS called P-BDSS aiming to address the *information overload* issue in business domain. As shown in Figure 5.1, the overall expectation is to ensure all the modules in P-BDSS can function as described in PVI, and together form a DS system.

- Collects plain-text business data from multiple sources.
- Original plain text data is processed by information extraction to generate structured information with good quality.
- The machine learning classifiers are able to determine the relevance and the sector of structured information.
- All extracted information is effectively presented by the state-of-the-art IV tools, and together with other communication and decision-support tools, provide useful decision support for BI users.

In the long run, we are working to improve the quality of IE and ML method to make P-BDSS achieve higher standards.

Overall, this thesis introduces novel methods of extracting information from unstructured data using document filtering, information extraction and supervised learning. Based on these methods, it presents a novel business DSS, based on lower-level details of research, forms a complete high-level architecture of the system. It demonstrates how different techniques

can be combined in one system to provide meaningful information in the business domain.

## 1.3 Thesis Structure

This thesis is organized as follows. Chapter 2 introduces a method for mining domain-specific patterns from text documents. It explains in detail how we use DF to collect and filter irrelevant documents, and how to extract the business activities from plain-text news. Chapter 3 introduces our experiments on supervised learning for multi-class, multi-label classification of business sectors of business activities in news articles. Chapter 4 presents research on interesting correlations between news, social media visibility and stock prices. Chapter 5 introduces the architecture, current status and the evaluation results of P-BDSS. Chapter 6 concludes the thesis.

# Chapter 2

# Acquisition of Domain-specific Patterns for Document Filtering

Document filtering aims to reduce the number of documents in a corpus required to be processed by information extraction while keeping the relevant documents. This chapter presents an automatic way to mine domain-specific patterns from text documents. It explains how these extracted patterns can be used for domain-specific document filtering. This chapter answers RQ1: How to filter irrelevant documents from many sources of continuously streaming news?

## 2.1 Data Collection

### 2.1.1 Introduction

We aim to collect as much relevant data as possible containing business activity events. This section introduces how we set up the data collection.

Our system received about 3000 business news articles daily from two partners in prior projects, who collect news from the Internet. One of them provided manually selected business articles and a summary for each article. This manual work introduced information delay ranging from hours to weeks. Another source provided business news collected using a web crawler, such as GNU Wget [17], GRUB [18], etc., and filtered using keywords. It contains the following two types of errors.

1. Errors of commission: a large number of irrelevant documents that contain the company name but not business activities, such as background information of a company; text containing "CEO of *company_name* says", "share this at facebook.com/*company_name*"; or a

media company reports something by putting its name in the text; etc.

2. Errors of omission: if we match both company name keywords and more business activity keywords like "acquisition", "launch" or "invest" to the query in order to increase the precision, this would, however, eliminate many relevant articles, which do not contain those specific keywords since the number of different terms or sentences to describe business activities in plain text is far too large to enumerate.

In order to develop a fully independent system, we started to collect raw data directly from Internet sources. In the later part of this research we moved to relying on data collected by our own system, rather than on data received from previous project partners. Initially, we use any news website providing business RSS feed in English. We also experiment to collect business news articles from sources of other kinds including company websites and news websites without RSS feed, etc.

### 2.1.2 RSS

Currently, P-BDSS stores 1282 news websites, which provide a RSS feed related to the business domain (e.g., BBC News Business, the New York Times Business Day, etc.). Some of these websites also provide RSS feed for non-business domains, such as politics, sports, arts, etc. P-BDSS also stores 106 of these non-business types of RSS sources. For each RSS feed, P-BDSS periodically (every half hour) checks the feed and fetches the new links of news if any. P-BDSS uses Readability [19] to eliminate unrelated data from the web link, such as navigation bar, images, links to related articles, etc., to extract the main article text from the page. The output information that P-BDSS stores in the database contains attributes of news articles, such as the plain text of the news article, headline, source URL, language and domain. We have been collecting news articles from RSS feeds since 2013 and by the time of this writing we have collected 3,849,414 English articles and 501,467 non-English articles. In this thesis, we call it "P-corpus".

### 2.1.3 Company Websites

We use Crunchbase and Freebase [20] to retrieve companies and their website URLs. P-BDSS then polls each of these URLs and tries to find if there is any news RSS link provided on the website. If the website has RSS, P-BDSS stores the RSS into the database and uses the same method as

described above to fetch the news articles and their attributes. If there is no RSS, P-BDSS tries to find what we call the "press link", such as "press/news/media" tab, etc., and extract the news from the inner links inside the press link. Some statistics of a complete new collection of articles from company websites in January, 2015 are show in Table 2.1.

| | **Crunchbase** | **Freebase** |
|---|---|---|
| Total company websites | 14989 | 24324 |
| Total company websites with RSS | 6790 | 8544 |
| Total articles | 73024 | 29189 |
| Total English articles | 60081 | 22421 |

Table 2.1: Company websites statistics.

## 2.2 Data Filtering

### 2.2.1 Problem

By manually checking 100 random documents collected from the business RSS feeds, we have found that even though we are using only business RSS feeds, 30% of them are non-business documents not containing any business activity/event. The non-business documents ratio goes up to 67% when we are checking 100 random documents collected from company websites. P-BDSS therefore aims to filter out irrelevant documents from the large number of documents collected.

### 2.2.2 Related Work

A large number of newspaper websites, which cover all domains globally, have been constructed to provide online daily news. Some of them gather related articles from a specific domain in order to serve specific interests. In the business area, Bloomberg [21], for example, monitors thousands of newspapers, magazines, trade journals, web sources and press releases for retrieving articles related to business information every day. Europe Media Monitor (EMM) [22] is another example, which gathers over 40,000 reports every day from news portals world-wide in 43 languages, classifies the articles, analyzes the news texts by extracting information from them, aggregates the information, issues alerts and produces intuitive visual presentations of the information found. The services provided by EMM is supported by document filtering (DF) and text categorization. Document

filtering (DF) is a technique often used before the IE process to retrieve relevant documents from a large dataset [23, 24]. Similar to DF, Text categorization (TC), which is used to classify news stories into different categories, is also a key technique for organizing large text datasets before the IE process [25]. While DF focuses on retrieving documents containing one specific type of information such as business activities and ignores other information that could also be important, TC tries to classify documents containing all kinds of important information into their own categories such as medical, business, security, etc.

P-BDSS experiments with two approaches to filter irrelevant documents. The first approach is to use supervised machine learning to build a classifier to decide whether the document is a business article. This is a TC task called "document routing" [26]. The second approach is to combine NLP techniques and frequent sequential pattern mining algorithm. With a minimal manual effort, we use these patterns to filter irrelevant non-business documents and generate domain-specific scenario-based document summaries at the same time.

### 2.2.3 Supervised Machine Learning Approach

TC categorizes documents into predefined categories. In P-BDSS, there are only two categories, i.e., business and non-business. P-BDSS uses only business documents for later information extraction; non-business documents are discarded. This task can therefore be solved by a binary classifier.

There are various types of statistical and machine learning algorithms suitable for this task. Typical ones include decision trees [23], K nearest neighbour [27], Naive Bayes, Support Vector Machines [23, 24], Neural networks, etc.

#### Data collection

We have a collection of over 3 million manually selected business news articles called "Biz-corpus" in this thesis. For each of these articles in Biz-corpus, we have identified at least one business activity. BIZ-corpus is used as the positive data for our task. In order to collect negative data, we have been using the 106 non-business RSS as described in Section 2.1.2 to collect non-business news articles in daily bases. From 23.03.2016 to 12.05.2016, we have collected 34,627 articles in a corpus called "Non-biz-corpus". They are used as negative data for relevance (business/non-business) classification.

### Training and Testing data

We randomly select 30,000 articles from both Biz-corpus and Non-biz-corpus for a supervised machine learning approach. We then split these 60,000 articles into 30,000 training and 30,000 testing data; each of them contains 15,000 Biz-corpus articles and 15,000 Non-biz-corpus articles.

### Data representation and learning algorithm

This approach uses a vector space model to represent articles. We use P-BDSS IE to pre-process these 60,000 articles into their vector representations. During the pre-processing, documents are split into sentences, which are split into words. Each word has a lemma and a part-of-speech (POS) tag. All the lemmas with certain types of POS tag [1] are selected as unigrams. Unigrams with a count of less than 7 times in the training data are removed. Then, we use these selected unigrams to generate consecutive bi-grams. The selected bi-grams need to appear at least 7 times in the training data. These selected unigrams and bi-grams are used as dimensions in the vector space to represent the articles. The experiments also adopt feature-selection methods to select the top 5000 unigrams and bi-grams as vectors, as ranked by Information Gain (IG) [28] defined as IG(f, C) = H(C) - H(C—f). According to our experiments, evaluation results do not improve when we have more than 5000 dimensions.

Support vector machine (SVM) is used for this experiments as it outperformed other commonly used algorithms in similar TC tasks [23, 29, 24]. We use standard "recall", "precision" and "F-measure',' as defined in the following formulas, to evaluate our experiments. In the formulas, TP is the number of correctly classified business articles. FN is the number of business articles that have been mis-classified as non-business ones. FP is the number of non-business articles that have been mis-classified as business ones. The evaluation results are shown in Table 2.2.

$$Recall = \frac{TP}{TP + FN} \tag{2.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

$$\mu\text{-}F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{2.3}$$

[1] We only use lemmas of the content words ( i.e., noun, adjective and verb) for our experiments.

| Class | Recall | Precision | F-measure |
|---|---|---|---|
| Business | 0.92 | 0.91 | 0.91 |
| Non-business | 0.91 | 0.91 | 0.91 |

Table 2.2: Evaluation results of machine learning approach to filtering.

### 2.2.4 Pattern-based Approach

In the recent decades, TC technology based on frequent patterns emerged [30, 31]. Instead of treating documents or sentences as a collection of independent words (vectors), the pattern-based approach considers the sentences as an ordered list of words. This approach then tries to find frequent patterns in the texts of the target corpus and use these patterns to classify the documents.

Our experiments using the pattern-based approach in PII are for two domains of news texts: medical epidemics and business intelligence. Section 2.3 explains our pattern-based method to extract frequent patterns in the business domain only. After quick manual post-selection, we can use these patterns to directly generate summaries, classify domain-specific documents, or induct them into the IE system. PII was mainly for finding patterns to generate a summary and to be used for IE, document classifying using these patterns is a byproduct.

## 2.3 Pattern Acquisition

P-BDSS uses several machine learning approaches to mine IE patterns [32, 33, 34]. One approach to pattern acquisition in PII is presented here. It contains three steps: 1) data collection and NLP pre-processing; 2) pattern mining; 3) scenario-based selection.

### 2.3.1 Data Collection and NLP Pre-processing

Unlike the machine learning approach presented above, a golden annotated text corpus for training is not needed, rather we use potentially relevant texts for acquisition of domain-specific patterns.

In the business domain, we use our data collection system from business RSS as described in section 2.1.2 to fetch potentially business-related news. Some example sentences from these articles are shown below:

- B1: The project involves a total investment of CNY 650mn (EUR 78.81mn).

- B2: Spanish bank, Banco Etcheverria, has approved a capital increase of EUR 499,659.12(US$ 660,886.19).

- B3: MPS sells stake in Biverbanca to CRAsti for €208.96mn.

We randomly selected 10,000 of these potentially relevant articles and performed NLP pre-processing. The sentences of these articles are divided into words; punctuation and stop words such as "an", "and", "he", "that" are removed. Each word in a sentence is treated as an item, and each sentence is treated as a sequence of items. The company is defined as a key item in the business area. The sentences that do not contain any company are considered irrelevant and are not used. We use a dictionary of company names (including their synonyms and acronyms) and a named entity recognition module in our IE system to determine whether the sentence contains a key item (company). In order to reduce sparseness, we replace companies and other general classification items with their types and use types as items. In this study, such items include:

- Country names are converted to "c-country-name".

- Company names are converted to "c-company-name".

- Items describing a human, such as "people", "man", etc., are converted into "c-human".

- Years are converted to "c-year".

- Numbers (e.g., "104", "1 280", "2,367" "five") are converted to "c-number".

- Currencies (e.g., "RMB", "€") are converted to "c-currency".

After this pre-processing, each transaction $T =< W_1, W_2, W_3, ..., W_n >$ describes a sentence mentioning at least one company. Each item W in T represents either a content word in the sentence, or a type of categorical item. These transactions contain at least one company name; example B1 is therefore removed. Transactions generated from our example sentences B2, B3 are shown below. The 10,000 articles generate 35,024 transactions.

- TB2: **c-country-name** bank **c-company-name** has approved capital increase of **c-currency c-number c-currency c-number**

- TB3: **c-company-name** sells stake in **c-company-name** to **c-company-name** for **c-currency c-number**

### 2.3.2 Pattern Mining

We use an Apriori-like algorithm [35] to find frequent sequential patterns, P, to describe business activities from these transactions, such as,

- P1: (c-company-name, sells, stakes, in, c-company-name)
- P2: (c-country-name, c-company-name, has, approved)

These patterns only find adjacent sequential items or types. This means if a transaction is formed by three sequential items $W_1, W_2$ and $W_3$, then the only allowed sequential patterns are $(W_1, W_2)$ or $(W_2, W_3)$ or $(W_1, W_2, W_3)$; pattern $(W_1, W_3)$ is not allowed. The support $S_p$ of each pattern $P$ is calculated as follows. The count $|P|$ is increased by one when P is found in a transaction $T$. If $P$ is found twice in the same transaction $T$, the count increases only by one.

$$S_p = \frac{|P|}{|T|} \tag{2.4}$$

$|T|$ is the total number of transactions. We set a minimum support $S_{min}$ to filter non-frequent sequential patterns with $S_p$ less than $S_{min}$.

The pipeline of this algorithm is described below.

- Initialize: read in the input and generate the initial counts for all two-item patterns, $(W_1, W_2)$.
- Iteration: starts with two-item patterns; stops when there is no possible next round patterns. Inside the loop, we calculate the counts for all possible frequent patterns generated by two frequent patterns in the previous round; and we use these counts to generate next round's possible frequent patterns. For example,
  - we have frequent patterns $(W_1, W_2)$, $(W_2, W_3)$ and $(W_4, W_5)$ initially;
  - we generate next round's possible candidates $(W_1, W_2, W_3)$ from $(W_1, W_2)$ and $(W_2, W_3)$ since the k-1 suffix of $(W_1, W_2)$ and k-1 prefix of $(W_2, W_3)$ are the same (where k refers to the number of items in the patterns);
  - if the suffix of one pattern is not the same as the prefix of another pattern, we do not generate any candidate from these two patterns, such as $(W_2, W_3)$ and $(W_4, W_5)$.
- Output all patterns where $S_p \geq S_{min}$.

Table 2.3 shows some statistics for different values of $S_{min}$.

| $S_{min}$ | # of rounds | | # of patterns | | # of key patterns | |
|---|---|---|---|---|---|---|
| | Med | Bus | Med | Bus | Med | Bus |
| 0.01 | 3 | 3 | 209 | 91 | 51 | 68 |
| 0.005 | 6 | 5 | 473 | 213 | 102 | 139 |
| 0.001 | 8 | 6 | 3159 | 2232 | 506 | 1280 |

Table 2.3: Acquired patterns for different $S_{min}$.

### 2.3.3 Scenario-based Selection

After acquiring frequent patterns, we manually select patterns that could be used to describe business activities. Through quick manual selection, we selected 259 patterns from those 1280 key patterns when $S_{min}$ is set to 0.001, which describe a business activity of a company. When selecting patterns, we also group them by assigning a scenario label to each pattern. Table 2.4 shows some examples of selected patterns, with their support and scenario label. In total, we have identified 12 frequent types of activities as scenarios in the business domain.

| Pattern | $S_p(\%)$ | Scenario |
|---|---|---|
| (c-company-name recall) | 0.21 | Product Recall |
| (c-company-name advertising) | 0.20 | Marketing |
| (c-company-name investments) | 0.19 | Investment |
| (c-company-name purchase c-company-name) | 0.13 | Acquisition |
| ( c-country-name c-company-name plans) | 0.23 | Planing |
| ( c-country-name c-company-name unveils) | 0.20 | New Product |
| ( c-country-name c-company-name opens) | 0.19 | Open |
| ( c-company-name contract is) | 0.18 | Contract |
| ( c-country-name c-company-name launch) | 0.14 | New Product |
| (c-company-name has launched c-company-name) | 0.13 | New Product |
| ( c-country-name c-company-name acquires) | 0.12 | Acquisition |
| ( c-country-name c-company-name appoints) | 0.12 | Management Succession |
| ( c-company-name deal is) | 0.12 | Contract |
| ( c-country-name c-company-name approves) | 0.11 | Announcement |
| ( c-country-name c-company-name buys) | 0.11 | Acquisition |
| ( c-country-name c-company-name supply) | 0.11 | Contract |
| (c-company-name is owned by c-company-name) | 0.10 | Ownership |
| (c-company-name has been awarded c-currency) | 0.11 | Investment |

Table 2.4: Examples of manually selected patterns for scenarios in the business domain.

## 2.4 Use of Patterns

We can use these extracted scenario-based patterns to perform several tasks.

The first task is to classify unseen documents into business or non-business categories to solve the document filtering problem as described in Section 2.2. An article is classified as a business article, if any transaction T of the article contains any of these 259 patterns.

The second task is to use the manually selected patterns to generate domain-specific scenario-based single-document summaries, in two steps. First, we use the same NLP pre-processing module as described in Section 2.3.1 to convert sentences of a document into transactions. Then, we select sentences that match any of the domain-specific patterns as summary sentences. A document containing no such sentence is regarded as irrelevant for the defined scenario in the domain.

Some statistics of generating summaries for 10,000 randomly selected documents are shown in Table 2.5.

| Domain | $\|Doc\|$ | $\|Sum\|$ | $Avg\|Doc_s\|$ | $Avg\|Sum_s\|$ |
|---|---|---|---|---|
| Business | 10,000 | 3,120 | 23.1336 | 4.78 |
| $\|Doc\|$: number of documents | | | | |
| $\|Sum\|$: number of documents that generate a summary | | | | |
| $Avg\|Doc_s\|$: average number of document sentences | | | | |
| $Avg\|Sum_s\|$: average number of summary sentences | | | | |

Table 2.5: Statistic results of summary evaluation.

We are also working on integrating the mined patterns into our IE system (Sector 5.3) for extracting attributes of pre-defined scenario events in the domain, such as company name, country, etc., as shown in Figure 5.2. Some of these patterns already match at least three categorical items. These categorical items can be directly converted into attributes in an IE output. For example, pattern "(c-company-name purchase c-country-name c-company-name)" can generate a business acquisition event with three attributes, i.e. the buyer company, the targeted company and the location of the acquisition.

## 2.5 Evaluation

To perform a manual evaluation for document filtering and summarization, we randomly select 20 documents that generate a summary and 20 documents that generate no summary. A business expert is invited to manually pick sentences from these 40 documents (T) to generate manual summaries. The expert knows 12 scenarios we are using and only selects sentences containing these 12 scenarios from a document to generate the summary.

The evaluation results are shown in Table 2.6, where $A_D$, $P_D$, $R_D$ and $F_D$ represent the accuracy, precision, recall and F-measure of the document-

level filtering while $A_S$, $P_S$, $R_S$ and $F_S$ represent the accuracy, precision, recall and F-measure of the document summarization.

| Document filtering | | | | Document Summarization | | | |
|---|---|---|---|---|---|---|---|
| $A_D$ | $P_D$ | $R_D$ | $F1_D$ | $A_S$ | $P_S$ | $R_S$ | $F1_S$ |
| 72.50 | 100.0 | 72.00 | 83.72 | 45.85 | 83.06 | 38.33 | 52.45 |

Table 2.6: Manual evaluation of summary.

The precision of both document filtering and document summarization are high. This demonstrates that our patterns are very reliable for scenario-based summarization in the business domain. When comparing the differences between summaries generated using our method and ones generated by an expert, we have found that sometimes the document describes exactly the same information in two sentences in slightly different ways, such as the title and the first sentence of the document. Our method selects both sentences because they both match the patterns, while the expert chooses one of them to generate the summary. This decreases the precision of the method. The recall of document filtering is much better than the recall of document summarization. This means that a relevant document will most likely describe the business activities using some frequent patterns at least in one sentence.

## 2.6 Chapter Summary

This chapter introduces P-BDSS data collection and filtering methods. It explains how P-BDSS collects and filters plain-text business news from over 1000 RSS sources.

Frequent Pattern acquisition is an important task for both DF and IE. This chapter presents one example of the machine learning approach to acquire patterns and demonstrates that a combination of NLP techniques and frequent sequential pattern mining algorithm can be used to extract patterns in a specific domain from unstructured natural-language text, i.e., news articles. These patterns can be used to filter non-business articles, generate business-specific scenario-based document summaries, and extract pre-defined scenario events in the business domain.

# Chapter 3

# Supervised Learning for Business Sectors

In this chapter, we present experiments of supervised learning classifiers in P-BDSS. These classifiers are used to obtain additional attributes of business activities that cannot be extracted by IE. This chapter answers the RQ2:How to extract the industry sector of the business activities from plain-text news?

## 3.1 Related Work

Supervised classification aims to predict the class labels of unseen instances using instances with known labels (labeled data). It uses labeled data and adapts various algorithms to build a model which takes instances (formed by a number of pre-defined features) as input and outputs the predicted labels of these instances.

Supervised classification is therefore well suited for our tasks, to extract the industry sectors (labels) of the business activities (instances) from plain-text news, since we have enough labeled data for both tasks.

The typical learning process can be divided into several steps [36]:

- Prepare data
- Choose an algorithm
- Fit a model
- Choose a validation method
- Examine fit and update until satisfied

- Use fitted model for prediction

P-BDSS follows the same work-flow. Both of our tasks are text categorization, which use text as instances and pre-defined categories as output. We use the vector space model (described in Section 2.2.3) in our experiments, which is a commonly used *data representation* for text categorization [37, 38, 39].

Text data is represented by a large number of distinct word types, which can exceed the number of training documents by an order of magnitude, [40]. Thus *feature reduction* becomes a key step in most text classification approaches. This aims not only to accelerate processing but also to improve categorization performance [37, 41] through avoidance of over-fitting, [42]. Reduction can be done either by *selection* of highly-relevant features or by *grouping* (i.e., clustering) features, [37]. In our experiments we use feature selection, which is based on comparing the discriminative power of a given word, relative to all other words in the feature set. Comparative studies of various feature selection methods can be found in, e.g., [28, 40].

We experiment with two supervised-learning algorithms for text classification [43]: Naive Bayes and Support Vector Machines (SVM) to compare the performances.

## 3.2 Sector Classifier

A sector classifier is built using supervised machine learning to decide the business sector (e.g, *Energy, Gas* or *Electronics, Telecommunications*, etc.) for any article analyzed by IE. We transform the multi-class, multi-label task as a set of binary sub-tasks. For each sector, we build one binary classifier. We explore several combinations of learning algorithms and feature selection methods, and evaluate them using a large amount of manually-labeled data.

It is traditional to assume the classifier will be applied in the future to data drawn from the same distribution as that, on which it was trained. However, the label distribution may change over time. For example, the industry-sector distribution of RCV1 corpus collected 15 years ago is unlikely to have similar sector distribution to ones in current Reuters news stream. In addition, the classifiers may need to label data from multiple news sources. Therefore, we focus on building a robust classifier, suitable for real-world classification, regardless of the distribution of labels. We therefore try to balance the training data for each sector to improve the performance of the whole classifier.

### 3.2.1 Data Collection and Representation

We use the publicly available Reuters corpus (RCV1)[1] for our experiments. RCV1 has 800,000 news articles, collected from Reuters 1996-1997. 351,810 articles of them have industry sector labels manually labelled by Reuters. There are seven- and five-digit codes for industry sector; seven-digit codes are children of the corresponding five-digit codes: e.g., *Fruit Growing* (I0100206), *Vegetable Growing* (I0100216) and *Soya Growing* (I0100223) are all children of *Horticulture* (I01002). We map all seven-digit codes to their corresponding parent codes, and merge labels that have the same name but different code.[2] 245 distinct sector labels remain after this pre-processing.

Each training and test document is represented using bag-of-words features. We use only nouns, adjectives, and verbs in our feature set, and remove all stop-words, proper names, locations, dates, and common verbs such as "have" and "do". We also generate bigrams that consist of these three parts of speech.

In total, 77,636 training instances (documents) have 49,262 unique features; each binary classifier has these 49,262 features initially before feature selection.

We use Information Gain (IG)[28] as described in Section 2.2.3 and Bi-Normal Separation (BNS)[40] to rank these features and select only the top 500 features.

### 3.2.2 Balanced Training and Testing Pools

We try to keep the training data as balanced as possible across sectors (as we discussed in Section 3.2), and ensure that the test set will contain a sufficient number of instances for every binary classifier in the array.

We rank all N sectors by size, and begin collecting data into the pools from the sector, $S_N$, that has the smallest number of instances in the corpus.

We randomly collect up to 600 documents labeled with $S_N$, and split them into two subsets: 3/4 for training and 1/4 for test. If there are not 600 documents ($< 600$) for $S_N$, all available instances are collected, with the same training/test proportion.

We then begin collecting data for the second smallest sector, $S_{N-1}$, and repeat the collection process, except now we first check how many documents labeled with $S_{N-1}$ are *already present* in the training and test

[1] http://about.reuters.com/researchandstandards/corpus/

[2] For example, we merge I64000 and I65000, both called *Retail Distribution*.

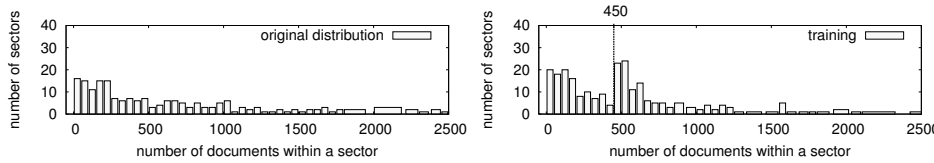



Figure 3.1: Document distribution among sectors in the training pool (right): aiming for approximately 450 documents per sector; distribution in the original corpus (left).

| **Sector** | **Instances** | **Sector** | **Instances** |
|---|---|---|---|
| Diversified Holding Companies | 3644 | Electricity Production | 1986 |
| Commercial Banking | 3153 | Agriculture | 1980 |
| Petroleum and Natural Gas | 2628 | Computer Systems and Software | 1805 |
| Telecommunications | 2145 | Air Transport | 1754 |
| Metal Ore Extraction | 2099 | Passenger Cars | 1713 |

Table 3.1: Number of *positive* instances in the training pool, for the ten most frequent sectors.

pools—which may happen due to multiple labeling (label overlap). We only collect missing ones to fill up 450 training and 150 testing data in total.

The collection process continues in this procedure for all sectors. Collection may be skipped for a sector if it already has more than 450 documents in the training pool (this happens for sectors with high label overlap). The resulting set, called the "balanced training data pool" has 77,636 documents. It is still skewed, as shown in Figure 3.1, on the right, although much more balanced than the initial distribution, shown on the left. In our experiments, we use only the 200 largest sectors, which cover approximately 99% of the original corpus.

Table 3.1 shows the most frequent sectors in the balanced training pool. We can see that although we only collected 450 positive training instances for *Diversified Holding Companies*, it still has 3644 positive instances in the pool, most of which were picked up when collecting data for other sectors.

In our experiments, we also use an *unbalanced* training pool, which is simply half of the corpus for comparison.

All data *outside* the balanced and unbalanced training pools—called the "test pool"—are available for the construction of test sets. From the test pool, we generate 10 samples of 10,000 documents each, using the original distribution in the corpus. We use one of these samples as a held-

out *development* set for parameter tuning, and the remaining nine as test sets.

### 3.2.3 Classification

We combine an array of binary classifiers for the multi-label classification task. Each classifier is trained for an individual sector. All classifiers in the array use exactly the same training set, where all documents labelled with a given sector are used as positive instances of the sector classifier, and all remaining training documents are used as negative instances. We try two kinds of supervised learning algorithms: Naive Bayes and Support Vector Machine (SVM), using the open source WEKA toolkit [44].

The SVM classifier outputs a binary decision for each document. For Naive Bayes, the output of each sector is a confidence score between 0.01 and 1; therefore, a decision threshold for classification is required. We learn the best thresholds in a series of thresholds (in increments of 0.01) by using the extended em development set. We then use the learning threshold to evaluate the remaining test sets.

#### Baseline rote classifier

We also set up a baseline classifier to compare the classification results. We use the IE to build a knowledge base, which contains sector distribution information for each company mentioned in the corpus. Then, we investigate how to use this information for text categorization. IE uses the named entity recognition (NER) module to find the company in the corpus. It distinguishes company names from other proper names in the text, such as people and locations. The NER module also incorporates variations of the same name, such as "Apple", "Apple Inc.", "Apple Computer, Inc." and so on.

The knowledge base contains the following many-to-many relations:

- document-sector
- document-company
- company-descriptor

We try to use various combinations of these relationships to build the rote classifier. We use the IE system to process the documentation from the training set, build the knowledge base, and then use that knowledge to sort the documents from the test set.

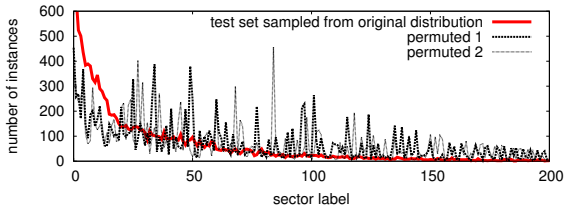



Figure 3.2: Label distributions of an original test set, and permuted test-sets (2 of 50 shown).

Table 3.2: Sector distribution for company "Apple".

| **Sector** | **Freq** | **Prob** |
|---|---|---|
| Computer Systems and Software | 549 | 0.61 |
| Electronic Active Components | 61 | 0.07 |
| Datacommunications and Networking | 36 | 0.04 |
| Telecommunications | 19 | 0.02 |
| Electrical and Electronic Engineering | 13 | 0.01 |

We assume that each company has its industry preference, that is, the industry that normally operates. As a result, the company name in the corpus co-occurs with specific sectors. For example, Table 3.2 shows the top sector that appears with "Apple"; it shows the number and proportion of the company's presence with the industry, i.e., the normalized count. As can be seen from the table, in 60 % of the cases, Apple appears in the documents labeled with the "computer system and software" sector, so it is natural to suggest that Apple belongs to this sector.

In fact, each document may belong to more than one sector, therefore, instead of choosing only the top-most frequent sector the classifier should return the entire sector distribution, which can be calculated using the evidence from all companies mentioned in the text. Thus the probability that document $D$ belongs to sector $S$, in the simplest case, can be defined by the formula:

$$P(S|D) = \frac{1}{|C_D|} \times \sum_{c \in C_D} P(S|c) \tag{3.1}$$

where $C_D$ is the set of companies mentioned in the document, and $P(S|c)$ is the proportion of times $c$ co-occurs with $S$ in the knowledge base; e.g.,

$$P(Computer\ Systems\ and\ Software|Apple) = 0.61 \tag{3.2}$$

This method would be reliable if the knowledge base contains sufficient evidence to associate the company with particular sector(s). Therefore, we only use companies that appear in the corpus three or more times. This means that if a document discusses a new (or little-known) company, the name-based classifier will be unable to find a sector for the document. In this case we can use descriptors to label the document, as descriptors allow us to use evidence gained from *other* companies in the corpus. For example, if company X is described in the text as "software company" we can assume that the sector distribution for this company would be similar to the sector distribution for "Apple". In this case the probability that document $D$ belongs to sector $S$ can be described by the formula:

$$P(S|D) = \frac{\sum_{c \in C_D} P(S|c) + \sum_{d \in d_D} P(S|d)}{|C_D| + |d_D|} \tag{3.3}$$

where $d_D$ is the set of all descriptors mentioned in the document. Note that $|C_D| \neq |d_D|$ because in this case we can use a company descriptor even when the company does not appear in any other document in the corpus.

This estimate of $P(S|c)$ based on co-occurrence may be inaccurate: for rare companies, some sectors may dominate the distribution by mere chance. Moreover, sector overlap may lead to a situation where the company belonging to one sector frequently co-occurs with another. Descriptors, therefore, may sometimes be more reliable for predicting the sector. To check this assumption, we define the probability that a company belongs to a particular sector as follows:

$$P(S|c) = \sum_{d \in d_C} P(d|C) \times P(S|d) \tag{3.4}$$

where $d_C$ is the set of all descriptors associated with company $c$ in the knowledge base. We then use (3.4) in (3.1) to obtain the final sector distribution for the document:

$$P(S|D) = \frac{1}{|C_D|} \times \sum_{c \in C_D} \sum_{d \in d_C} P(d|C) \times P(S|d) \tag{3.5}$$

Note that in this case the company name is substituted by a set of descriptors; however it is possible to use the company name in combination with company descriptors:

$$P(S|D) = \frac{\sum_{c \in C_D} \sum_{d \in d_C} P(d|C) \times P(S|d) + \sum_{c \in C_D} P(S|c)}{2 \times |C_D|} \tag{3.6}$$

**Combined Classifiers**

We experiment with several methods of combining the Rote classifier with the balanced probabilistic classifiers to see whether the combination can produce better *overall* prediction of the sector labels. One method of combining is a simple two-stage process: for each document, we first try to identify sectors using the Rote classifier; if that does not return any sectors, we then attempt to classify using the statistical classifiers. We also experiment with the reverse order of these classification stages. The motivation for this method is to give the overall system a "second chance" at classification, in the hope that together the two methods may overcome their respective shortcomings. Another method of combining classifiers is to return the *union* of the results of the two classifiers—rote and probabilistic. We learn the optimal threshold for each classifier in the combination using the development set.

### 3.2.4 Evaluation Results

Common measures in text classification, i.e., precision, recall, and F-measure are adopted. In evaluating multi-label classification, *macro-averaging* and *micro-averaging* are commonly reported [45, 46]. In micro-average evaluation, first the numbers of true- and false-positives, and true- and false-negatives are counted for all instances in the test set, and then the standard measures, e.g., recall or precision, are calculated using these numbers:

$$Rec_\mu = \frac{\sum_{i \in S} TP_i}{\sum_{i \in S} (TP_i + FN_i)} \qquad Prec_\mu = \frac{\sum_{i \in S} TP_i}{\sum_{i \in S} (TP_i + FP_i)} \tag{3.7}$$

$$\tag{3.8}$$

$$\mu\text{-}F1 = \frac{2 \times Rec_\mu \times Prec_\mu}{Rec_\mu + Prec_\mu} \tag{3.9}$$

where $S$ is the set of all classes. In the macro-average evaluation scheme, the measures are calculated for each class *separately* first, and then these are averaged across all classes:

$$Rec_M = \frac{\sum_{i \in S} Rec_i}{|S|} \tag{3.10}$$

$$Prec_M = \frac{\sum_{i \in S} Prec_i}{|S|} \tag{3.11}$$

$$M\text{-}F1 = \frac{\sum_{i \in S} F1_c}{|S|} \tag{3.12}$$

We report both evaluation schemes, although we focus more on the macro-average scores, as explained below, since they are less dependent on the particular distribution of labels in the corpus. We denote the macro-averaged F-measure by M-F1, and micro-averaged F-measure by $\mu$-F1.

**Comparison of classifiers and feature selection methods**

Evaluation results are shown in Table 3.3. As seen in the table, the rote classifier that uses company names and descriptors from the document (**name+desc**) yields the highest F-measure among single classifiers. The SVM classifier yields higher performance than NB, independently of the feature selection method used. IG performs better than BNS with both Naive Bayes and SVM. Combining rote classifier using company names and descriptors with SVM+IG yields the best overall performance. To save space we show only selected classifier combinations in Table 3.3; it can be seen in the table that the classifiers that have higher scores alone work better in combination, and that, for combined classification, taking the union of classified sectors gives better results than the two-stage method.

There are five papers directly comparable to our work in that they use a large number of sector labels and report micro- and/or macro-averaged F-measures: [47, 48, 49, 50, 51]. In Table 3.4 we present a detailed comparison between their results on RCV1 industry labels and ours. It can be seen in the table that the difference between M-F1 and $\mu$-F1 for our classifiers is smaller than that reported in prior work. This supports the claim that classifiers trained on balanced data are less sensitive to changes in label distribution—which is one of our main objectives.

## 3.3 Chapter Summary

This chapter explains how ML classifier is built and how the classifier extracts sectors of business activities, from plain-text to supplement the structured business activities extracted by DF and IE.

We first present experiments with supervised learning for labeling business news documents with multiple industry sectors. We explore several combinations of learning algorithms and feature selection methods, and evaluate them using a large amount of manually-labeled data. The main contribution of this experiments is that combining a named-entity-based rote classifiers with the balanced classifiers yields better results than either classifier alone. This method outperforms the best M-F1 previously reported, while using considerably less training data to built the classifier.

| | *M-average* | | | *μ-average* | | |
|---|---|---|---|---|---|---|
| **Classifier** | **Rec** | **Pre** | **F1** | **Rec** | **Pre** | **F1** |
| *Statistical classifiers* | | | | | | |
| **NB+IG** | 31.3±0.9 | 21.9±0.6 | 19.7±0.6 | 31.5±0.5 | 22.4±0.6 | 26.2±0.5 |
| **NB+BNS** | 34.2±1.1 | 16.6±0.6 | 15.8±0.5 | **33.1**±0.7 | 13.4±0.4 | 19.0±0.5 |
| **SVM+IG** | 31.9±1.3 | **59.2**±1.1 | **37.1**±1.2 | 30.5±0.4 | **72.7**±0.6 | **42.9**±0.4 |
| **SVM+BNS** | **32.7**±0.9 | 55.2±1.0 | 36.2±0.7 | 30.1±0.5 | 70.8±0.6 | 42.2±0.5 |
| *Rote classifiers* | | | | | | |
| **name** | 36.8±0.8 | **65.2**±1.0 | 44.5±0.7 | 45.9±0.5 | **60.5**±0.4 | 52.2±0.5 |
| **descriptor** | 8.8±0.3 | 38.4±1.2 | 11.6±0.3 | 16.4±0.2 | 29.0±0.3 | 20.9±0.4 |
| **name+desc** | **39.4**±0.8 | 63.3±0.7 | **46.2**±0.7 | **48.5**±0.5 | 57.8±0.5 | **52.8**±0.4 |
| **name⇝desc** | 11.9±0.2 | 48.0±0.9 | 16.0±0.3 | 20.6±0.4 | 39.0±0.4 | 27.0±0.4 |
| **name+name⇝desc** | 39.2±0.8 | 60.0±0.8 | 44.8±0.6 | **48.5**±0.5 | 54.5±0.4 | 51.3±0.4 |
| *Combined classifiers* | | | | | | |
| **name→SVM+IG** | 46.2±1.0 | **73.7**±0.8 | 55.1±0.8 | 52.5±0.5 | **75.9**±0.4 | 62.0±0.4 |
| **SVM+IG→name** | 47.0±1.2 | 67.7±0.9 | 53.7±1.1 | 49.9±0.3 | 73.9±0.3 | 59.6±0.3 |
| **name ∪ SVM+IG** | 52.2±1.1 | 66.3±0.8 | 56.9±0.9 | 57.7±0.4 | 71.1±0.3 | **63.7**±0.4 |
| **name+desc→SVM+IG** | 48.4±1.1 | 69.2±0.7 | 55.5±0.9 | 56.2±0.5 | 70.0±0.3 | 62.4±0.4 |
| **SVM+IG→name+desc** | 46.7±1.0 | 70.2±0.8 | 54.6±0.8 | 53.8±0.5 | 71.2±0.4 | 61.3±0.4 |
| **name+desc ∪ SVM+IG** | **53.7**±1.0 | 64.5±0.8 | **57.2**±0.8 | **59.7**±0.4 | 68.1±0.3 | 63.6±0.3 |

Table 3.3: Results from all classifiers and feature selection methods, averaged across 9 test sets randomly sampled from original distribution. For each classifier, the best threshold is trained on one random, originally-distributed development set. Rote classifier names correspond to the following formulae from Section 3.2.3: **name** – (3.1), **name+desc** – (3.3), **name⇝desc** – (3.5), **name+name⇝desc** – (3.6). For combined classifiers → and ∪ denote the two-stage and union combining methods, respectively (Section 3.2.3).

The predicted results are stored in the database with the articles and their structured business activities. The information is visualized in DSS as described in Section 5.5.

| **Reference** | **Algorithm** | M-**F1** | $\mu$-**F1** |
|---|---|---|---|
| [48] | SVM | 29.7 | 51.3 |
| [51] | SVM | 30.1 | 52.0 |
| [49] | SVM + re-ranking | 34.1 | 62.8 |
| [50] | Naive Bayes | - | 70.5 |
| [47] | Bloom Filters | 47.8 | **72.4** |
| Our work: **name+desc** $\cup$ **SVM+IG** | | **57.7** | 63.8 |

Table 3.4: Classification results on RCV1 industry sectors, compared with state of the art.

# Chapter 4

# Joint Analysis of News and Social Media

This Chapter explains how to combine the extracted information with external data using EDM and presents a study for exploring interesting correlations between news, social media visibility and stock prices. By integrating these external data, P-BDSS aims to provide deeper decision support to users. This chapter answers RQ3: How to link external information (e.g., social media visibility and stock prices) with news?

## 4.1 Related Work

The relationship between traditional news, social networks, and stock prices is interesting for business analysts, Web scientists, data journalists, etc.

Previously, a co-analysis of news and social media has been studied [52, 53, 54, 55]. There are two interrelated goals from these studies: to find information that is complementary to the information found in the news, and to control the amount of data that needs to be downloaded from social media. [56] studied Wikipedia page views; they demonstrated that some events, such as earthquakes or terrorist attacks, triggered bursts in Wikipedia hits.

Recent research also shows that there is a certain correlation between stock prices and news if the news are properly classified [15, 16]. For example, paper [15] demonstrated that sentiment analysis can be useful for stock prediction: negative sentiment (pessimism) predicts temporary decreases in returns on market [16] demonstrated that Information Extraction technology can be extremely useful for stocks prediction, since various types of events affect prices in different manners. For example, deals and partner-

ship announcements tend to be very positive while legal announcements tend to be negative.

There are also attempts to correlate stock prices with social media activities. For example, paper [57] demonstrated that both the number of posts and the interaction strength among users correlates with stock price changes. Investors' special social media, e.g., SeekingAlpha and StockTwits are compared and analyzed in this paper [58]; they show that some authoritative users can well predict the stock price changes.

## 4.2 Experiment Setup

We first investigate how companies and products mentioned in the news are portrayed on different social media. In this thesis we have focused on Wikipedia and Twitter.

Currently, we focus on numerical measurement and analysis of the content. We count the number of Twitter posts where the name of the company occurs. Tweets are short Twitter posts, where usually a user shares her/his impression about an entity (company, product or person), or posts a related link. Wikipedia, on the other hand, is used for obtaining more in-depth background information about a company or a person.

### 4.2.1 Twitter Data

There are some technical limitations of collection for Twitter data. While data can be collected through the Twitter API in near-real time, the API returns posts only from recent history (7-10 days). This means that keyword extraction and data collection should be done relatively soon after an entity appears in the news. Another limit is that one API key can call a limited number of requests within a time window.

We use "New Product" events extracted by our IE system to construct queries to Twitter API. One query contains the company name and product name, which are the slots of a "New Product" event. Everyday, we extract about 50 "New Product" events from news articles and generate 50 queries to Twitter API. The returned results of each query are the tweets that contain both company name and product name in the query.

While company name and product name are used as keywords in each query, other slots of the "New Product" event are used for analysing the results of the query. These slots, which include industry sector, country, product descriptor and report date of the event, are used to label the returned tweets from the query. For example, we extract an event "Nokia launches Lumia 928 in US" and retrieve 2,000 tweets that contain both

Table 4.1: Dataset description.

| Time | Events | Tweets |
|---|---|---|
| Nov 2012 - May 2013 | 1764 | 3,842,148 |

"Nokia" and "Lumia 928". Since the event is related to the industry sector "Telecommunications" decided by our sector classifier (described in Chapter 3) and the country "US", we consider these 2,000 tweets are also related to "Telecommunications" and "US". Then, we can group all returned tweets by industry sectors or country and analyse these tweets.

The dataset is described in Table 4.1. We started the experiment in November 2012 and the results seen in this thesis include the data till May 2013. In total there are 1764 different products and close to 4 million tweets.

### 4.2.2 Wikipedia Requests

We collect a complete Wikipedia page request history for all editions starting from early 2008, updated daily. We can then use this collection to retrieve the daily hit count history for any Wikipedia article. Mapping a name of an entity to a Wikipedia article can not always be done, but the mapping appears to be quite robust in the vast majority of cases. Thus, we have used the Wikipedia data to explore and demonstrate visibility in social media in the results presented in the following section.

Covering multiple sources is important due to the different nature of the social media. One of our next steps is to check whether integrating additional services, (e.g., YouTube) will add value to the system; we also plan to analyze the content of the tweet texts, and try to determine, e.g., whether the sentiment of the posts is positive or negative. This can then be correlated with upward vs. downward stock fluctuations.

## 4.3 Results

We present three types of results: A. Most frequently tweeted company news, B. visual analysis of correspondence between Wikipedia views, news hits and stock prices, and C. time-series correlations between news hits and Wikipedia views.

### 4.3.1 Most Frequently Tweeted Company News

The majority of 1140 companies in our experiments appear in one business event only; less than 10 tweets have been found for 50% of them. The most frequently tweeted companies are presented in Table 4.2. It presents the number of events for a company in our dataset, the maximal number of tweets per one event and the total number of tweets for this company.

We can see that only events related to well-known IT companies, e.g., Facebook, Google and Microsoft, produce more than 100,000 tweets. Apple, which is the fourth frequently tweeted company, produces almost 10 times less tweets than Microsoft. Other companies presented in Table 4.2 are telecommunication and automotive companies, food and drink producers, cosmetics and cloths suppliers.

### 4.3.2 Visual Analysis of Correspondence

We chose three companies—Alstom, Malaysia Airlines, and General Motors in this experiments. In order to demonstrate the visual correspondence, we present the number of news documents, the number of views of English-language Wikipedia page, and stock data of these companies, using data from March to December 2014.

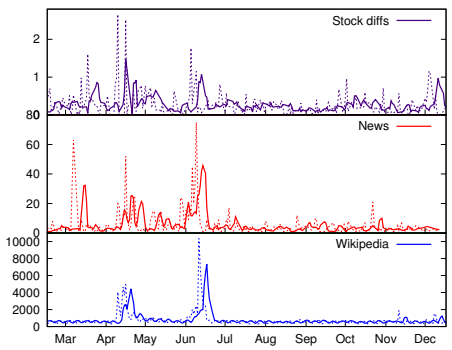
In Figure 4.1, the top plot shows the daily *difference* in stock price—the absolute value of the opening price on a given day minus the price on the previous day, obtained from Yahoo! Finance. The middle plot shows the number of company news. The bottom plot shows the number of hits on the company's Wikipedia page. In each plot, the dashed line represents the daily values and the bold line is the value smoothed over three days.

Figure 4.1a shows the data for the French multinational Alstom. Alstrom is known for its train-, power-, and energy-related products and services. We can see a pattern where the stock price and news mentions seem to correlate rather closely. There is also some correlation between Wikipedia page hits and other plots. The news plot shows three major spikes, with two spikes in Wikipedia hits. The March peak corresponds to news about business events (investments), whereas the other peaks are related to political issues, which could trigger more activity in social media. For example, "The French government bought 20% of Alstom shares in June" caused an active public discussion in social media.

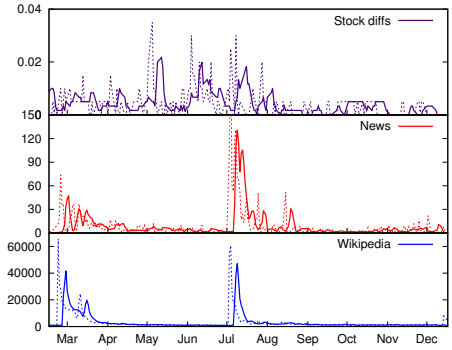
From the example of Malaysia Airlines, we can see a strong correlation between news mentions and Wikipedia hits in Figure 4.1b. Two major spikes are the two severe incidents of Malaysia Airlines in 2014. On March 8, they lost one aircraft, and on July 17 another was shot down in Eastern

Table 4.2: Most frequently tweeted companies.

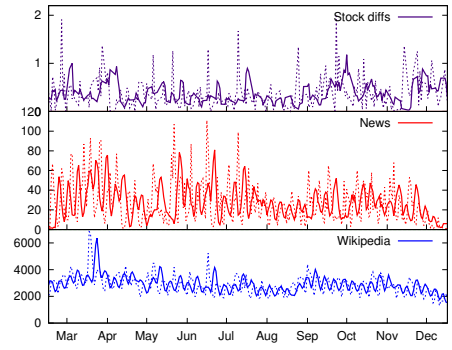
| **COMPANY** | **# events** | **max # tweets** | **total # tweets** |
|---|---|---|---|
| Facebook | 10 | 444188 | 1464226 |
| Google | 14 | 105054 | 410272 |
| Microsoft | 9 | 169086 | 285059 |
| Apple | 4 | 19619 | 34054 |
| Lego | 2 | 15371 | 26001 |
| Audi | 3 | 13373 | 13829 |
| T-Mobile | 2 | 7884 | 9043 |
| Huawei | 3 | 8559 | 8561 |
| Acer | 1 | 6099 | 6099 |
| Coca-Cola | 6 | 2432 | 5891 |
| Samsung Electronics | 19 | 1966 | 4578 |
| Volkswagen | 1 | 4454 | 4454 |
| Netflix | 1 | 4369 | 4369 |
| Starbucks | 1 | 3993 | 3993 |
| Lenovo | 1 | 3129 | 3129 |
| Nokia | 2 | 2760 | 2793 |
| Sony | 4 | 2404 | 2705 |
| Seat | 1 | 2641 | 2641 |
| Walmart | 1 | 2575 | 2575 |
| Telefonica | 2 | 2065 | 2085 |
| Tesla Motors | 1 | 2082 | 2082 |
| H&M | 2 | 1787 | 1787 |
| Land Rover | 2 | 1668 | 1668 |
| Adobe | 2 | 1507 | 1571 |
| Dell | 1 | 1074 | 1074 |
| Lacoste | 2 | 799 | 821 |
| Adidas | 5 | 728 | 784 |
| Flipboard | 2 | 628 | 637 |
| Pringles | 1 | 572 | 572 |
| Skoda | 2 | 447 | 523 |
| Mazda | 1 | 511 | 511 |
| Chanel | 2 | 436 | 436 |
| Chobani | 1 | 414 | 414 |
| Marvel Comics | 1 | 354 | 354 |
| Electronic Arts | 2 | 261 | 348 |
| Protno-M | 1 | 330 | 330 |
| Puma | 1 | 297 | 297 |
| Oracle | 2 | 193 | 295 |
| Reddit | 1 | 293 | 293 |




(a) Alstom




(b) Malaysia Airlines




(c) General Motors

Figure 4.1: Daily differences in stock prices, number of mentions in news and number of Wikipedia hits in 2014 for three companies.

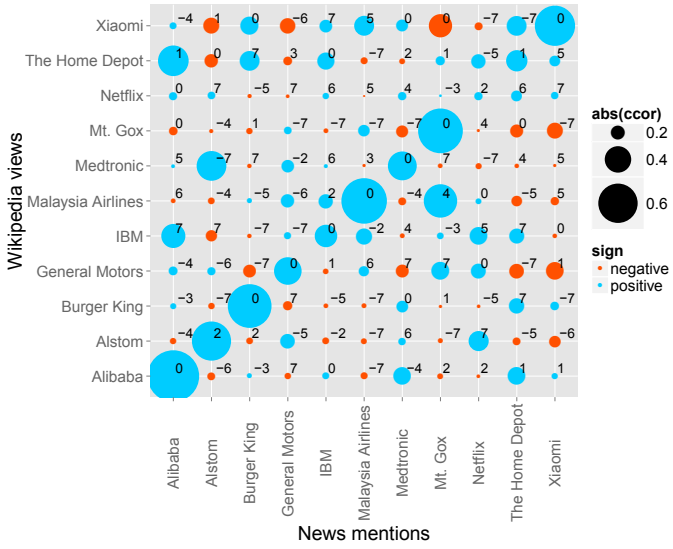
Ukraine. The correlation with the stock price is less obvious.

Figure 4.1c demonstrates the correspondence data for General Motors, which was affected by numerous product recalls throughout the year. General Motors appears in the news and has been looked up on Wikipedia throughout the covered period. The stock price also dropped over the entire year.

Even though most of the local oscillations are due to normal fluctuations in the weekly flow of data on the Internet, some broader-range correspondence is still detectable from the plots.

### 4.3.3 Time-series Correlations

In this third experiment, we select eleven big companies from different in-




Circle width represents strength of correlation; color represents sign of correlation: blue is positive, red is negative; the numbers indicate the time lag (in days) at which the correlation with the greatest magnitude is obtained for the given pair: positive lag means that Wikipedia views follow news mentions.

Figure 4.2: Cross-correlation between Wikipedia views and news mentions for 11 companies.

dustry sectors, including Alibaba, Alstom, Burger King, General Motors, IBM, Malaysia Airlines, Medtronic, Mt. Gox, Netflix, The Home Depot, and Xiaomi. For each of them we collect two time series: daily news mentions and Wikipedia views from March to December 2014. We calculate the cross-correlation between all possible pairs in this dataset, for a total of 121 cross-correlations[1]. We limit the lag between time series by seven days, based on the assumption that if there exists a connection between news and Wikipedia views it should be visible within a week.

The results are presented in Figure 4.2, where the circle size represents correlation strength, the colors indicate correlation size: blue means positive correlation, red means negative; the numbers mean the time lag at which the highest correlation for a given company pair was obtained: positive lag means that Wikipedia views followed news mentions, negative lag means that news followed Wikipedia views. We can see that the largest correlations and the lowest lags can be found on the diagonal, i.e., between news mention for a company and the number of views of the company Wikipedia page. There are two exceptions: The Home Depot and Netflix among the 11 companies. For Netflix, news mentions and Wikipedia views

[1]We use standard R ccp function to calculate cross-correlation.

do not seem to be strongly correlated with any time series. News about Alibaba show a surprising correlation with Wikipedia hits on Home Depot on the following day. At present we do not see a clear explanation for these phenomena; these can be accidental, or may indicate some hidden connections (they are both major on-line retailers).

## 4.4 Chapter Summary

In this chapter, we have presented three experiments of the interplay between company news, social media visibility, and stock prices. Attributes of extracted business activities is used to construct queries to various social media platforms. The study results demonstrate the utility of collecting and comparing data from a variety of sources.

We are able to discover interesting correlations between the mentions of a company in the news and the views of its page in Wikipedia. The correspondence with stock prices was less obvious. These interesting correlations can be presented to the end user in P-BDSS. Users may investigate which news events correspond to bursts in social media visibility and, conversely, find possible causes for unusual social media activity. We expect that combining both kinds of information would be useful for business professionals, Web scientists, and researchers from other fields.

# Chapter 5

# PULS Business Decision-Support System (P-BDSS)

From previous chapters, we can see that this research is focusing on document filtering, Information Extraction and machine learning, and how these techniques can be adopted for processing unstructured data for various purposes. Other supporting tools including information visualization for user interfaces, information storage, parallel and distributed computing are also investigated and used. Based on these research topics, this chapter presents the design and architecture of a complete DSS, which aims to extract, analyze and organize information from plain-text news and deliver it to users in a concise and easy-to-access form. This chapter answers RQ4: How to construct the business DSS using information from unstructured text data?

## 5.1 Architecture

The conceptual architecture of the P-BDSS is shown in Figure 5.1. It consists of:

- Document Filtering module (DFM)
- P-BDSS NLP engine
  - Information extraction module (IEM)
  - Machine learning module (MLM)
- Data storage module (DSM) and external data module (EDM) running on the back-end

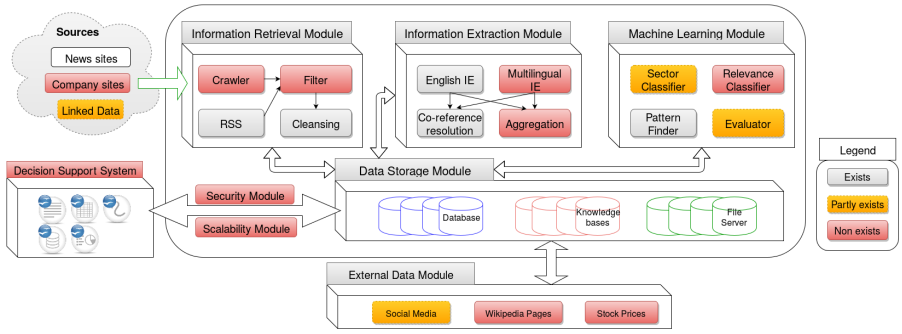



Figure 5.1: Static Structure Diagram of P-BDSS.

- DSS on the front-end

P-BDSS conceptual workflow can be divided into three sub-parts:

1. Collect raw data from possible sources: (DFM, DSM)

2. Process through P-BDSS natural language processing (NLP) pipeline: (IEM, MLM, DSM)

3. Visualize the intelligence and interact with the user: (DSS, DSM, EDM)

## 5.2 Document Filtering Module

Before this research work, PULS receives about 5000 business news articles from our partner system, which introduces the information delay. The coverage of these news articles also depends solely on the partner system.

P-BDSS develops its own document filtering module that collects raw data directly from Internet. Initially, the targeted sources is any business news website providing RSS feed in English. We also experiment to collect raw business data as much as possible from multiple sources of various kinds including company websites and news websites in multiple languages. DF described in Chapter 2 is used to filter out irrelevant documents from a massive number of documents available on-line in real time. Currently, DFM obtains unstructured raw data from 1124 different types of sources on the Web (e.g., BBC news business, Nytimes Business Day, etc.).

## 5.3 Information Extraction Module

### 5.3.1 Background

IE is an important technology behind the system. It extracts pre-defined structured information from unstructured plain text. The structured information can be stored directly into the database for later query. Starting from 1987, IE was spurred by a number of Message Understanding Conferences that were sponsored by DARPA and organized by the US Naval Ocean System Center [59]. After MUC-7, IE research has been stimulated by the *Automatic Content Extraction* (ACE) evaluations, whose objective is to develop technology to support processing of text from a variety of sources (e.g., newswire, broadcast conversation, etc.) [60]. Along with ACE evaluations, a number of IE systems have been developed for different purposes in a variety of domains. Here are some examples:

- In 2000, WhizBang! Labs launched its first product, a job search website called *FlipDog* [61]. It used the IE technology from WhizBang! Labs [62] that spidered corporate websites for job postings, extracted the data, categorized the jobs extracted, and added them to FlipDog's database.

- At New York University, Grishman, Huttunen and Yangarber developed an extraction system *Proteus BIO* [63] that gathered news from multiple sources, and updated the extracted information to a database. The information extracted was infectious disease outbreaks containing disease name, location, date, and other attributes of the outbreak event. Patwardhan and Riloff [64], Phillips and Riloff [65] also worked on IE systems in the disease outbreak domain.

- In the biomedical and molecular biology domain, a large number of IE applications were developed in recent years for different purposes. While some applications covered the full process of text mining from filtering related documents to mining the biomedical interactions and relations, some of them only provided useful tools for one or more stages of the full process [66]. Protein-protein interactions extracted from the scientific manuscripts were another focus in this domain [67, 68, 69].

- A multilingual security-related event extraction system [70] was developed using IE in 2013 in order to track cross-border criminal activities.

- Recently, business researchers published several approaches to applying language technologies to the financial domain. E.g., an IE system has been developed to process Dow Jones Newswire, and was shown to help in predicting stock prices [16]; a system for detection of economic events is described in [71]; domain-specific polarity dictionaries were used in [72, 73] to determine financial constrains of companies mentioned in official reports. Most of these approaches use rule-based and dictionary-based techniques. Several publications describe application of machine-learning techniques to the business domain; e.g., for learning extraction patterns [74] and supervised text classification [75, 76]. An overview of language analysis applications in the business domain can be found in the recent survey in [1].

Before this research, the extracted information by PULS for business domain includes business activities (events) and their associated entities or slots (e.g., country, location, company, company descriptor[1], person, product, product descriptor)[2]. How to increase the quantity and improve the quality of these extraction results are the key research focuses of P-BDSS (as described in Section 5.3.2 and Section 5.3.3).

### 5.3.2 Methods of Extraction

An extraction pattern contains an indication for specific variable tokens and their surrounding context. While the surrounding context is fixed, the tokens are variable. For instance, "A B recalled by C D on E" is a simple pattern. It can be used for matching recall event, as shown in Figure 5.2 from the sentence "A total of 717,950 vehicles have been recalled by US-based General Motors (GM) on 23 July 2014 ...". The pattern provides the fixed context in this example; A, B, C, D and E are the variable tokens. According to the definition of required slots by the IE system, A is an integer, B could be a noun group representing any product, C is a noun group representing any descriptor of a company (e.g., car maker, energy company, etc.), D could be a company or an organization name, E may be a date.

An IE system usually has a large number of such extraction patterns to match required facts. Different systems, according to their purposes and domains, may have quite different patterns. Finding, building and

[1] A company descriptor is a phrase that best defines what a company is doing, such as mobile phone maker, car seller, oil company, etc.

[2] Depending on the type of activities, the slots are different. For example, an investment business activity has investor (a person or company), target (a company or product or other targets), country, value, time, etc.

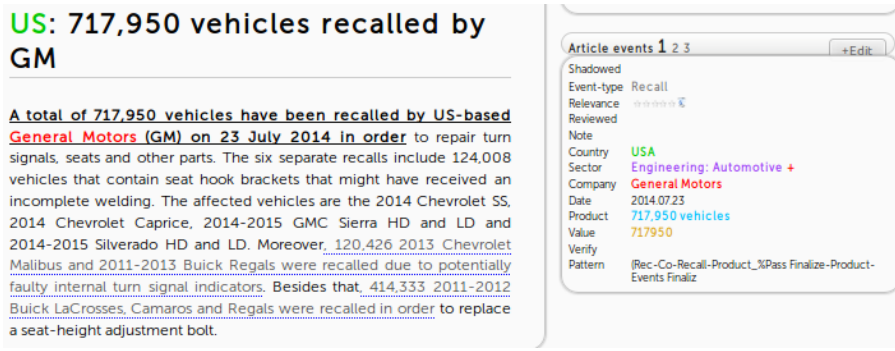



Figure 5.2: P-BDSS IEM example.

fine-tuning the extraction patterns is therefore the core task of building P-BDSS IEM since the quality of the resulting template largely depends on the quality and coverage of these patterns. In general, there are two different ways of obtaining suitable patterns.

- **Knowledge Engineering Approach**, using this approach, linguists and knowledge experts of the required domain manually define the extraction patterns [4, 5, 6, 7, 8].
  - **Advantages**: the precision of facts extracted by these manually defined patterns is normally high.
  - **Disadvantages**: this approach requires a lot of time to be spent on creating and evaluating the suitable patterns; it is usually domain- and language-specific; the recall of the facts could be low since it is difficult to come up with all possible patterns that can match all kinds of natural language descriptions of the fact especially manually.
- **Machine Learning Approach**, by using machine learning algorithms, this method identifies the essential regularities for Information Extraction from a training set that consists of suitable annotated texts, and these essential regularities are then used for creating extraction patterns. Example IE systems using this approach include *AutoSlog-TS* [9, 10, 11], *CRYSTAL* [77, 78], *PALKA* [79], etc.
  - **Advantages**: the recall of facts extracted by using these acquired essential regularities is normally high especially when the

training set is good enough in terms of size, regularity, domain specificity, etc.; it is usually much faster than the manual approach to obtain a large set of useful patterns; theoretically it is domain and knowledge independent.

– **Disadvantages**: while obtaining the useful patterns quickly, this approach also brings a number of irrelevant patterns that could dramatically decrease the precision of facts; finding good training and testing sets of text is also a challenging task since bad training and testing set results in bad patterns.

In IEM, P-BDSS uses both knowledge engineering and machine learning approaches. Frequent pattern mining, which is one of the machine learning approaches P-BDSS could use is presented in Section 2.3. For each new domain and scenario, we first use the machine learning approach to help us quickly acquire a number of possible extraction patterns. We then invite our linguistics and domain experts to manually select appropriate ones from them and we integrate these patterns into our IEM. We use these patterns and other modules in IEM as described in Section 5.3.3 to extract predefined structured information from plain-text news articles. Currently, we are extracting pre-defined structured information from 16 business scenarios (event types) as shown in Table 5.1.

| Scenario | Scenario |
|---|---|
| investment | new product |
| management succession | contract |
| acquisition | ownership |
| closing | order |
| layoff | marketing |
| bankruptcy | merger |
| product recall | accident |
| funding rounds | others [3] |

Table 5.1: Business scenarios.

Besides pattern-based extraction, statistical methods have been introduced in recent years since the extraction patterns were found to be too brittle for more noisy unstructured sources. Two kinds of techniques were deployed in parallel: *generative models based on Hidden Markov Models (HMM)* [80, 81, 82, 83] and *conditional models based on maximum entropy* [84, 85, 86, 87, 88]. Although the statistical methods for extraction is newer, there is no clear winner until now. The characteristics of two

methods can be summarized as follows,

- **Pattern-based methods**
  - driven by hard predicates.
  - more useful in specific domains where human involvement is both essential and available.
- **Statistical methods**
  - decision made by weighted sum of predicate firings.
  - more robust to noisy data.

### 5.3.3 P-BDSS IEM Structure

P-BDSS IEM is mainly constructed by the following 5 modules:

1. Pre-processor, which segments text into headlines, paragraphs, sentences, tokens, and does part-of-speech tagging.
2. Shallow parser, which identifies and groups sequences of lexical items into lower-level structures, e.g., noun phrases, verb groups, appositions, etc. In the example in Figure 5.2 the parser found that "a land-mark ice-class Arctic rig" is a noun group that is used as a unit in later stages of the analysis. An important phase in this process is named entity recognition (NER), which finds proper names ("Keppel", "Chukchi Sea").
3. Pattern matcher, which gives semantic interpretation to lower-level fragments and combines them into events.
4. Co-reference resolution or discourse processing, which identifies and links mentions of the same entity through the text, and fills missing event slots.
5. Output generator, which produces results in the form suitable for storing in the database and later querying and downstream analysis.

The resulting output consists of items as slot values of a structured template (Figure 5.2). Based on the linguistic analysis, the output produced by components 1 to 2, pattern matching is used also in P-BDSS to extract facts. These facts are then used to fill the slots of the resulting template.

Same events from the same document are merged. We are also experimenting to merge same events from different documents based on the key attributes of the events, such as company name, location, event type, sector, etc.

## 5.4 Machine Learning Module

We experiment with two kinds of learning classifiers built on machine learning in P-BDSS MLM. The utility or relevance of an event is essential for providing users with better decision support. According to the criteria described in papers [89, 90], a business activity event, described in today's news is much more relevant than an event that happened one year ago and is described in today's news as background information. When giving such an old event a lower relevance score, people can more easily focus on the surveillance of the current business world by filtering all low-relevance events.

We also develop a sector classifier. For each event from these 16 scenarios as shown in Table 5.2, the sector classifier decides the top-level business sectors (e.g, *Energy* or *Electronics*, etc.) and 388 more specific second-level sectors (e.g., *Energy, Gas*, *Electronics, Telecommunications*, etc.) under these top-level sectors. [4] We treat the multi-class, multi-label problem as a set of binary sub-tasks, with one binary classifier for each sector. We explore several combinations of learning algorithms and feature selection methods, and evaluate them using a large amount of manually-labeled data. Further, we focus on building robust classifiers, suitable for real-world classifications—rather than on improving the performance on a single, static corpus—by balancing the data given to each classifier during training. The detailed experiment is presented in Chapter 3.

| Sector | Sector | Sector |
|---|---|---|
| Agriculture | Chemicals | Commodities |
| Construction | Consumers | Corporate |
| Drinks | Education | Electronics |
| Energy | Engineering | Environment |
| Finance | Food | Forest |
| Government | Health | Labour-Market |
| Materials | Media | Mining |
| Politics | Services | Social-Issues |
| Trade | Transport | |

Table 5.2: Business top-level sectors.

[4] We use two-tiered system, each top-level sector contains a number of more specific second-level sectors (e.g., Top level sector ").

## 5.5 Decision-support System

Using IEM and MLM, P-BDSS gathers a large number of business information aiming to provide useful decision support to DSS users. By selecting graphical elements and interactive methods according to the data using state-of-the-art IV technologies[5], we are developing a useful DSS which is able to provide access to P-BDSS extensive and profound business information, assist users to navigate the decision-support result more effectively, and also provide powerful tools for decision makers to communicate with DSS and other decision makers.

### 5.5.1 Related Work

Now, we have the structured business activities with attributes that are extracted by the DFM, IEM and MLM described in previous sections. These activities are stored in the database. How to use these facts effectively brings us the topics "Information Visualization (IV)" and "Decision Support (DS)". In this section, principles of these topics and how these principles are taken into account when creating a decision support system are introduced and investigated.

#### Information Visualization

IE technologies have enabled the collection of a large amount of information from a variety of resources in a short time. Rapidly growing data also imposes a challenging task for presenting them effectively. Data summarized and presented in a suitable and illustrative manner would demonstrate and reveal valuable and significant facts to users, while an inadequate representation confuses users and dramatically decreases their enthusiasm to explore potentially very useful information.

Information Visualization includes all developments and progresses made in *Data Visualization*, *Infographic*, *Scientific Visualization* and *Visualization Design*. It is committed to creating an intuitive way to convey abstract information in order to assist people in *using their vision to think* [95]. By utilizing the advantage of the human eye's broad bandwidth pathway into the mind, all kinds of visual representations and interaction techniques enable users to see, explore or understand a massive amount of information immediately.

In short, Information Visualization provides access to extensive and profound knowledge. Through investigating and analyzing the characteris-

[5]Google maps Api [91], Highcharts [92], BMVIS [93], SIMILE Widgets [94], etc.

tics of the data, an appropriate design of visualization using IV techniques would dramatically demonstrate more potential value of abstract data. We will come back to see how IV is applied in DSS in the following section and the rest of the thesis.

### Decision Support System

Decision support systems (DSS) is a computer-based information system which assists users to make decisions by utilizing the data, models, knowledge and human-computer interactions provided by the system [12]. It is a more advanced outcome generated by the development of Management Information System (MIS). Aiming at improving the quality of decision making, DSS provides an interactive environment and useful tools to decision-makers for compiling useful information from various information resources, analyzing the problem, modeling and simulating the decision-making processes towards the final solutions.

**Structure of DSS** The Dialog-Data-Modeling (DDM) architecture proposed by Sprague and Carlson is accepted by most academics as the initial structure of DSS [96, 97]. They describe that DSS has three fundamental components [12]: the *data module*, the *model module* and the *user interface module*. When DSS was combined with Expert System (ES), the *inference module* was also added into DSS:

- **Data module** contains the database and its Database Management System (DMS) [98]. The DSS database contains a large number of internal information (such as internal accounting data), or external data (such as financial indices). These raw data would need to be gathered and extracted to the data format suitable for decision-makers to manage, analyze, update and retrieve [12].

- **Model module** includes the Model Base (MB) and its management system (MBMS). MBMS integrates various decision-making models to analyze the internal and external information from the database. An example model could be the mathematical model that analyzes and simulates a complex problem, comes up with feasible solutions and help the user to choose a solution from those options. MBMS also includes the modeling language for users to customize the models or build their own models [99].The basic abilities of MBMS include [100]:

  1. satisfying the users' needs for different models.
  2. capability of integrating model and data.

3. providing an easy-to-use interface to communicate with the models.
4. capability of sharing models.

- **User interface module** is the interactive part of the DSS. It accepts and inspects the user requests, calls the functional components within the system to invoke the model runs, data analysis and knowledge inferences to effectively solve the decision problem [101]. *User interface module* has three main actors: the user, computer hardware and software systems. The communications between human and DSS can be divided into three parts [96]:

  1. **The Action Language**: refers to any way the user would use to communicate with the DSS, such as keyboard, mouse and any other controlling hardware of software instructions.
  2. **Display or Presentation Language**: refers to the DSS output of information in any form that the user can explore, such as monitor, printer, etc.
  3. **User Documentation**: contains any required knowledge that the user needs to know to use the DSS effectively, such as user manuals.

- **Inference module** is formed by the KB, Knowledge Base Management System (KBMS) and the inference engine component. In order to solve many unstructured or semi-structured problems, specialized knowledge is required besides the standard features of DSS. So in the modern DSS, the KBMS is also an important sub-system in addition to DBMS, MBMS and DGMS [99].

The advantages of the graphical format for displaying information make Information Visualization best serve the displaying and presenting purpose of DSS output in the user interface module. The more advanced interactive IV enables the user to navigate the decision support result more effectively and also provides a much easier tool for decision makers to communicate with DSS. Almost all DSS integrate interactive IV in its user interface module. The most commonly used graphical displays are *line chart*, *bar chart* and *pie chart*. More advanced graphical tools such as *dashboard*, *relational graph* or combined graphical formats are also popular. By selecting suitable formats from these options according to the DSS output, DSS simplifies the profound scientific questions into a visual image, which gives full play to the human cognitive ability. Not only does DSS with IV facilitate

the researchers to study and analyse the problems, but it also provides a powerful tool for them to communicate with it to acquire more powerful decision support. In the following sections, we will see how IV is applied into P-BDSS DSS.

### 5.5.2 Visualization

#### Unit views

Currently, P-BDSS experiments with several ways to visualize the structured information, including table, list, document, timeline, graph and rank as described below.

**Table view** *Table view* displays the structured business activities in table format. Each row is a business activity (e.g. investment, new product launching, etc.), while each column represents one attribute of the business activities, such as company name, location, sector, etc. It supports sorting by any column, Regular Expression and advanced searching as shown by Figure 5.3.

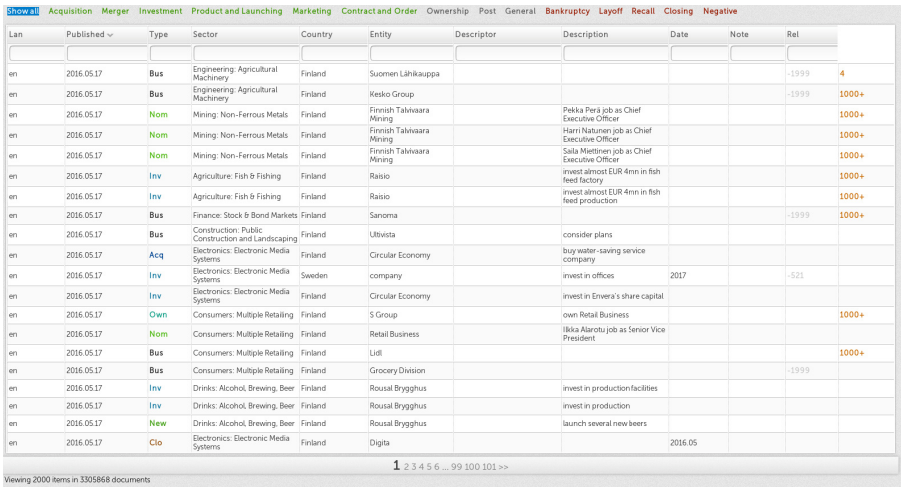



Figure 5.3: Table view.

**List/box view** *List view* displays business activities in an ordered list. Each box is an item in the list, representing a business activity. It contains

the company name, the location of the activity, activity type, sectors, and the sentence containing the business activity. The list is in chronological order by default, and can also be ordered by company name, distance between the activity location and user's location, relevance of the activity, etc. (Figure 5.4).

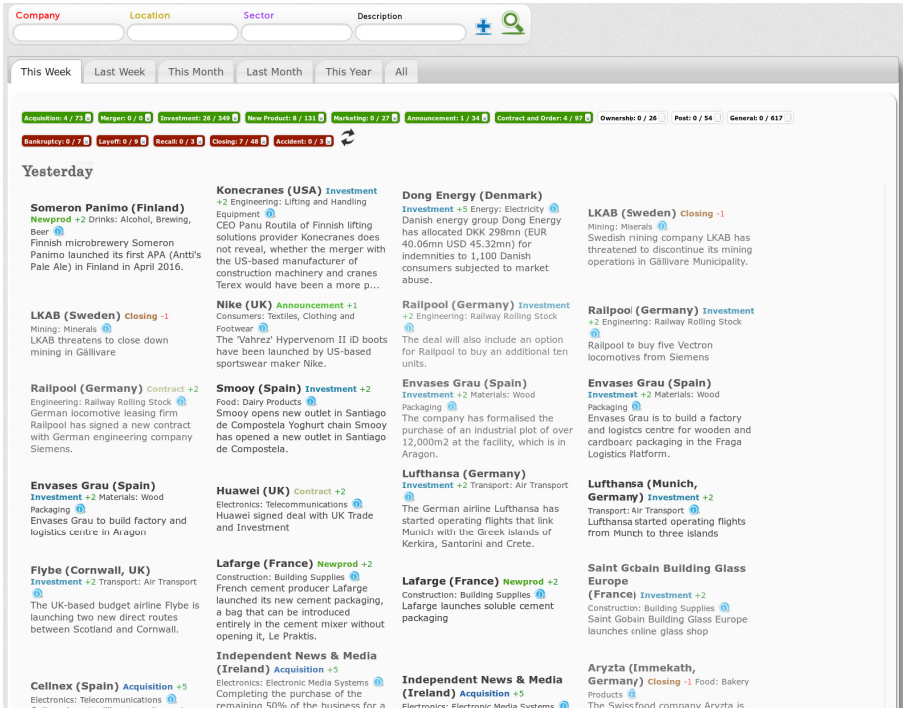



Figure 5.4: List view.

**Document view** *Document view* displays the news article and all extracted business activities. The sentences containing a business event are highlighted to help the user locate the activities in text. It allows users to make comments to, add, edit or rate the relevance of any activity event. It also provides possible related events links as shown in Figure 5.5.

**Timeline view** *Timeline view* displays activity events chronologically. It best demonstrates the frequency of activities during a certain period. Figure 5.6 shows an example of events in the *Electronics, Telecommunications*

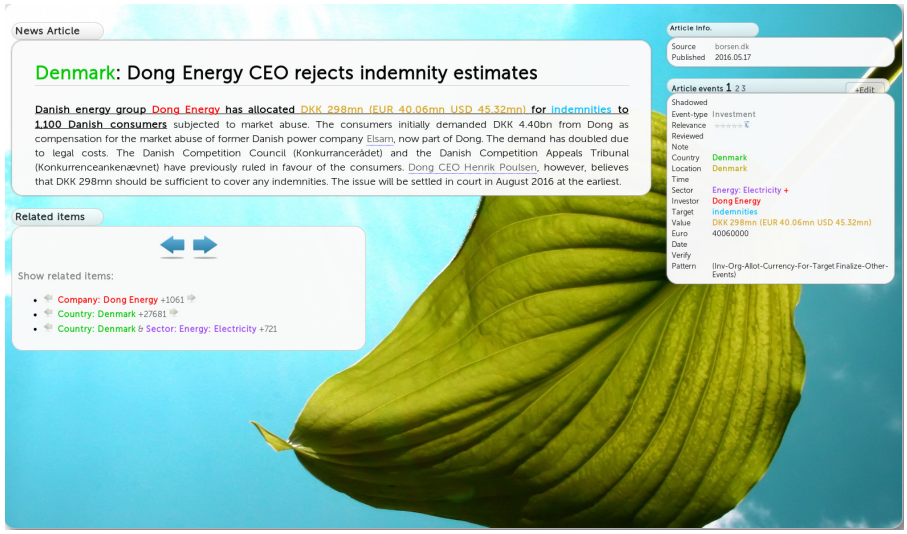



Figure 5.5: Document view.

*Terminals, Telephones* sector displayed in the *timeline view*.

**Graph** *Graph view* (Figure 5.7) displays the network among attributes of business activities. Currently, we are using three types of entities, i.e., company, person and product to construct the graph. Each node of the graph represents one instance of these entities, and each edge represents the relationship of two node entities that IEM extracts from the text. For example, an acquisition event "*Company A* acquires *Company B*" will produce two nodes "Company A" and "Company B" and one directed edge of type "acquire" from "Company A" to "Company B" in the graph. Another example "*Company C* hires *Person A* who was the CEO of *Company B*" will generate another two nodes "Company C" and "Person A" in the graph, and two more edges of type "employ" from "Company C" and "Company B" to "Person A". With millions of such extracted business activities as nodes and edges in the graph, users may generate the following kinds of graphs easily:

- Business network between entities, e.g., between Nokia and Apple.
- Business network around a certain entity, e.g., up-to 3 jumps /footnote3 jumps means the minimum path between two entities are 3

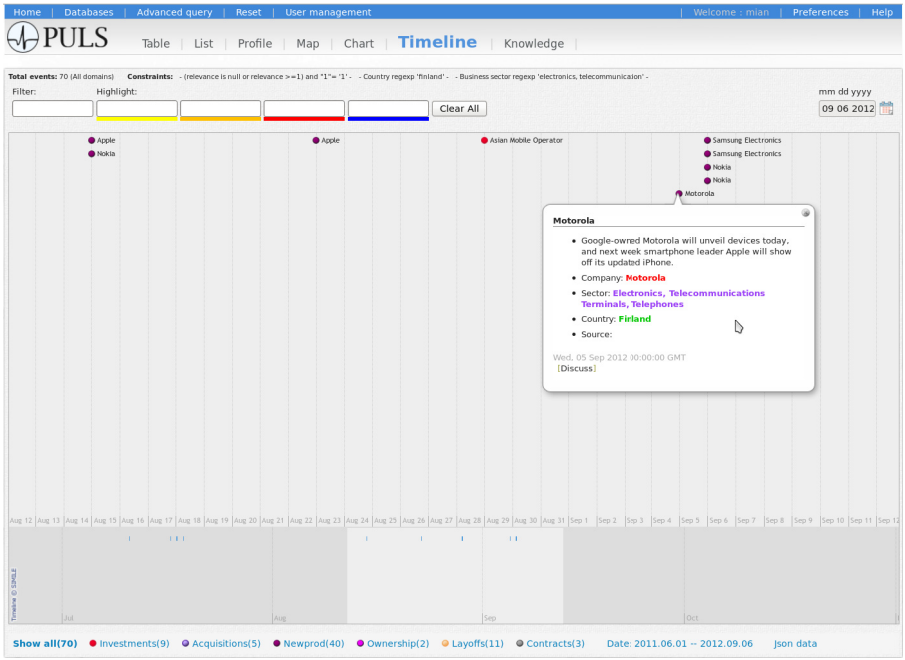



Figure 5.6: Timeline view.

edges of a company, such as Alicorp (Figure 5.7).

- All subsidiaries of a company.
- All companies a person works for.
- All Companies selling/producing a certain type of products, e.g., car.

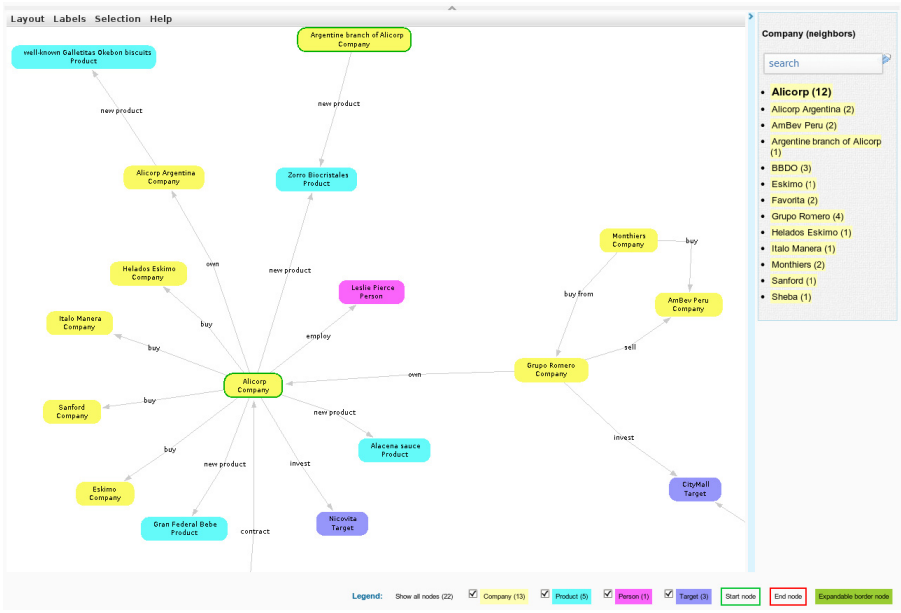



Figure 5.7: Graph view of Alicorp.

**Rank** *Rank view* displays companies or sectors in an ordered list for a certain period. As shown in Table 5.1, each activity is assigned a positive or negative score depending on the sentiment polarity of the events. We then sum up scores for all events of a certain company or sector in a given period and use the sum value to sort the list. An example of the rank view of companies for the current month is shown in Figure 5.8. In this view, we can select the period; sort the list by positive, negative or total score; search the rank of companies in a certain sector, or go to list/table view to check the activities of a company or sector in the period.

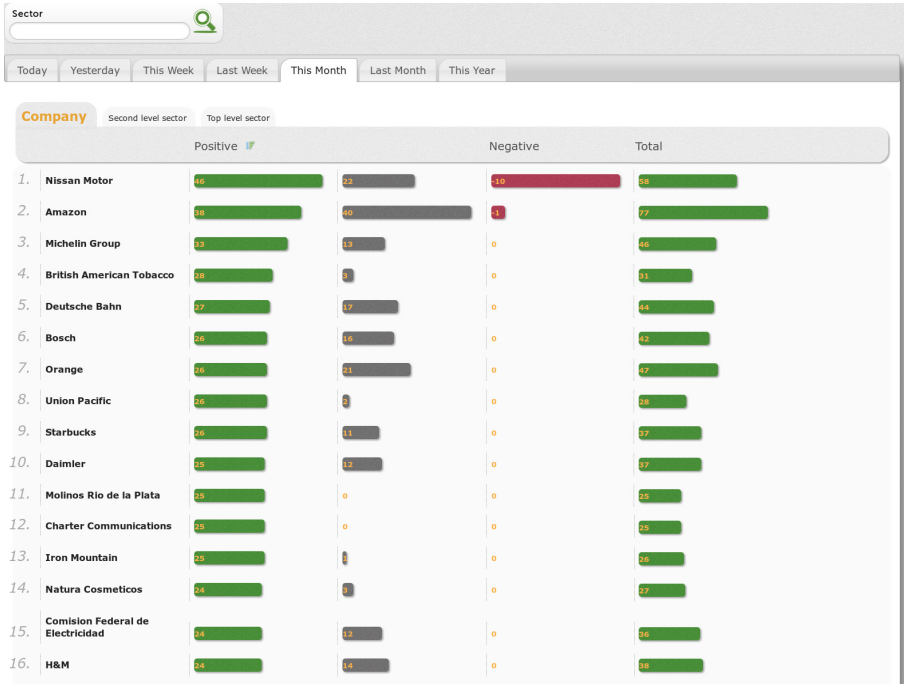



Figure 5.8: Rank view of companies in current month.

### Compound views

Besides these single types of views, P-BDSS also experiments with two kinds of compound views.

**Profile** *Profile view* is formed by a number of different components. Each component in the profile view displays a certain type of information related to a profiling attribute, e.g., a company or a sector. For example, Figure 5.9 is the profile view for Nokia. In this view, we can view the following types of information:

- Descriptor of Nokia: IEM extracts the descriptor of Nokia from plain-text. A descriptor is the piece of information that describes a company. For example, "Mobile phone maker", "Telecommunication company" are typical descriptors used in the news to describe Nokia. In profile view, we display the most frequently used descriptors with probability above a threshold.

- Sectors of Nokia: for each activity that involves Nokia, MLM classify the sectors of the activities. We then store each company to a sector pair relationship in our knowledge base. In the profile view, we display the 5 most frequent sectors of Nokia from these pairs.

- Rank of sentiment polarity of Nokia as described in rank view above.

- Related companies of Nokia: we use frequency distribution of descriptors and countries to compute the similarity score (Similarity AB) of two companies (as shown in formulas 5.1 - 5.3). We then display 50 companies which have the biggest similarity score with Nokia in a cloud view. The bigger the score, the bigger the company name is displayed.

- Geography frequency distribution of Nokia in a pie chart.

- Event type frequency distribution of Nokia in a pie chart.

- A list of persons employed/fired by Nokia.

- A list of new products launched into market by Nokia.

$$F_{d \in D} = \frac{|d|}{|D|} \tag{5.1}$$

$$F_{c \in C} = \frac{|c|}{|C|} \tag{5.2}$$

$$Similarity_{AB} = \sqrt{\sum_{d \in D_A d \in D_B} Min(F_{Ad}, F_{Bd}) \sum_{c \in C_A d \in C_B} Min(F_{Ac}, F_{Bc})} \tag{5.3}$$

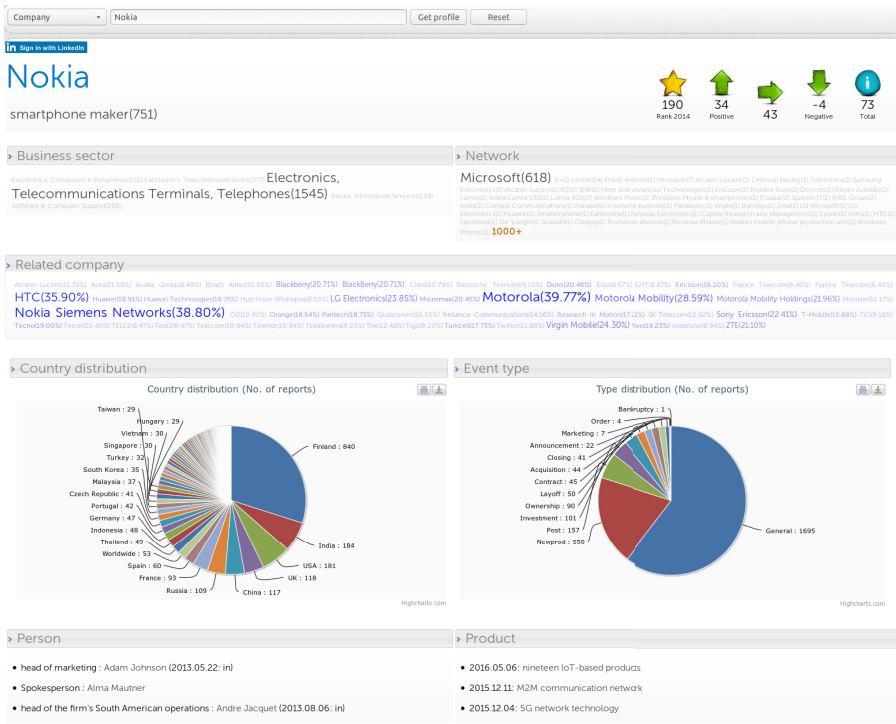



Figure 5.9: Profile view of Nokia.

**Dashboard** *Dashboard view* has a similar look as profile view. Instead of displaying related information using different IV elements for a particular attribute, Dashboard view displays information boxes that are customised by the users for a certain period, such as today. For example, the user may select 2 sectors and 2 companies to follow, this will generate 4 information boxes in dashboard view, each box contains the overview information

of the sector or company today. Sector overview includes the following information that is gathered today:

- Rank polarity of this sector as described in rank view above.
- Total events of this sector.
- Big players that have the highest ranks.
- Top stories of this sector.
- Company distribution pie chart.
- Geography distribution pie chart.

Company overview includes the following information of what happened today:

- Rank polarity of this company as described in rank view above.
- Total events of this company.
- Related companies or persons.
- Top stories of this company.
- Sector distribution pie chart.
- Geography distribution pie chart.

All above views are also interconnected with each other. For example, any time users see a company or sector, they can go to the profile view of this company or sector; when users see a business activity in any view, they can go to the document view of this activity, etc.

## 5.6 Low-level Supporting Tools

DSM handles the storage and retrieval of various kinds of data in P-BDSS. Raw information gathered by DFM is stored using MongoDB [102], which adopts document/object-based storage model[6] and other information is stored in relational databases.

Since the number of data introduced in this research is vast, parallel and distributed computing have been used in all the modules on the back-end in P-BDSS to save time. For each task, a manager is to be developed for monitoring the distributed tasks and delivering the final combined results.

[6]File systems based on object-based-storage model: Google File System (GFS) [103], MogileFS [104], Facebook's Haystack [105], etc. It is more suitable for managing large-scale library where file is written once, read often, never modified and rarely deleted [105].

## 5.7 Evaluation

The main task of P-BDSS is to maximize the quality of extraction results, similarly to the general objective of other IE systems. The term *quality* here varies according to the extraction purpose and the evaluation metric. We continuously perform several types of evaluations on P-BDSS IE systems ranging from the formal MUC-ACE style evaluations that measure the quality of the IE system results in terms of recall and precision for every slot in the template, to more coarse-grained evaluations that concentrate on a set of key slots in the template such as *company name*, *country*, *date*.

### 5.7.1 Evaluation Setup

The first step is to **create testing key file and generate result file**. We manually gather a testing corpus. Each document in the corpus may or may not contain an appropriate event. From the corpus, we create a testing key file that contains a number of templates with correct slots manually extracted from the documents in the corpus. Each template represents an event in a document. We then use the corresponding P-BDSS IE subsystem to process the same corpus, extract events from the documents as templates and append the templates into one result file. Suppose the corpus has 100 documents, both the key file and result file would have 100 such sets of templates. The result file could have a different number of templates for the same document.

The next step is to **compare all the templates between the key file and result file**. We build a script for comparing the key and system output every time when something changes in the P-BDSS IEM (e.g., new patterns introduced). Since the number of extracted templates in a document could be different from the number in the answer key, the script must find the most probably matching templates within a document between key and result. Further, the script is configurable to choose the slots for evaluation. For the formal MUC-ACE style evaluation, we compare every slot in the template, i.e., all the slots as shown above. Suppose the key file contains 200 templates, it will then have 200 *N.key* country slots that need to be filled. If the P-BDSS IEM fills 120 *N.correct[country]* slots and 120 *N.incorrect[country]* slots (including unfilled slots that should be filled), then, we can see that, *N.correct[country]* + *N.incorrect[country]* can be larger than 200 since P-BDSS IEM may extract some additional wrong events. Recall, Precision and F-measure of country slots are computed using formulas 5.4 - 5.6.

$$Recall[country] = \frac{N.correct[country]}{N.key[country]} = \frac{120}{200} = 0.6 \tag{5.4}$$

$$Precision[country] = \frac{N.correct[country]}{N.correct[country] + N.incorrect[country]} = 0.5 \tag{5.5}$$

$$F - measure[country] = \frac{2 \times Recall[country] \times Precision[country]}{Recall[country] + Precision[country]} = 0.545 \tag{5.6}$$

The overall results are calculated using *N.correct* and *N.incorrect* for all the slots we want to evaluate as shown in formulas 5.7 - 5.10,

$$N.*[all] = N.*[country] + N.*[company] + N.*[value] + N.*[time] + ... \tag{5.7}$$

$$Recall[all] = \frac{N.correct[all]}{N.key[all]} \tag{5.8}$$

$$Precision[all] = \frac{N.correct[all]}{N.correct[all] + N.incorrect[all]} \tag{5.9}$$

$$F - measure[all] = \frac{2 \times Recall[all] \times Precision[all]}{Recall[all] + Precision[all]} \tag{5.10}$$

*F-measure[all]* is the final evaluation result of P-BDSS IEM. We therefore try to balance the precision and recall to reach a higher F-measure score.

### 5.7.2 Evaluation Results

The key file currently contains 231 documents, and 2188 slots. The evaluation result of IEM is shown in Table 5.3. The evaluation result of the sector classifier in MLM has been presented in Chapter 3.

## 5.8 Chapter Summary

This chapter introduces the architecture, pipeline and overall evaluation of P-BDSS, which collects raw plain-text data from the Internet, extracts the

| **Event type** | **# keys** | **R** | **P** | **F1** |
|---|---|---|---|---|
| nomination | 83 | 100.0 | 78.0 | 87.6 |
| orders | 29 | 86.0 | 69.0 | 76.6 |
| acquisition | 52 | 81.0 | 75.0 | 77.9 |
| investment | 69 | 75.0 | 52.0 | 61.4 |
| product launch | 24 | 79.0 | 68.0 | 73.1 |
| contract | 72 | 76.0 | 60.0 | 67.1 |
| all events | 383 | 83.8 | 63.2 | 72.1 |
| **Slot in event** | **# keys** | **R** | **P** | **F1** |
| company | 521 | 61.2 | 50.0 | 55.0 |
| country | 293 | 70.0 | 44.3 | 54.2 |
| all slots | 2118 | 64.0 | 43.0 | 51.4 |

Table 5.3: P-BDSS evaluation results.

structured business information from raw data and builds BI tools based on the structured information.

The overall objective of this research is to ensure that all the modules in the system can function well, and together form a stand-alone DS system to address the information overload issue in business domain. Currently:

- Plain-text business news are gathered by DFM from over 1000 news sources in English.
- Plain-text news are processed by IEM to produce structured information (business activities) from 16 scenarios with good quality. In the long run, we are adding extraction patterns to improve the quality of IE results in terms of both recall and precision.
- MLM is extracting more information for the structured information, i.e. multiple sectors and relevance of the structured information. Details are presented in Chapter 3.
- All of these types of information are presented by state-of-the-art IV tools in DSS. We are planning to evaluate the usability with real BI users.

# Chapter 6

# Conclusion

The on-going information explosion affects in particular the business domain, in relation to corporate strategy and business decisions. The ability of users to leverage information and being aware of market trends is essential for day-to-day functioning. This thesis presents research on a natural language processing system, which aims to address the problem of information overload in the business domain. Overall, this thesis introduces novel methods for extracting information from unstructured data in various ways, including document filtering, information extraction, and supervised and semi-supervised learning. Based on these methods, it presents a novel business DS system, from lower-level details of research, gradually forms a complete high-level architecture of the system. It demonstrates how different technologies can be combined in one system to provide meaningful information in the business domain.

The main research question is to study "How to address information overload and provide decision support in the business domain?" In order to answer this question, this thesis focuses on the following sub-questions:

- RQ1: How to filter irrelevant documents from many sources of continuously streaming news?
- RQ2: How to extract the information not explicitly presented in text from plain-text news?
- RQ3: How to link external information, e.g., social media visibility and stock prices with news?
- RQ4: How to construct the business DSS using information from unstructured text data?

This thesis contains six scientific articles as presented in Table 1.1 to answer these four questions.

Chapter 2 presents paper PI to answer RQ1. PI presents experiments on a pattern-based classifier to filter non-business news articles from a live business news corpus, in order to improve the quality of the corpus. This business news corpus is collected automatically by a document collection module as described in section 2.1. By manually checking 100 random documents collected from news sites, we have found that even when we are only using business RSS feed, 30% of them are non-business documents not containing any business activity. The non-business documents ratio goes up to 67% when we are checking 100 random documents collected from company websites. These non-business documents dramatically decrease the precision of the extraction result of IE. We therefore need to filter out non-business documents from a massive number of documents collected to improve the precision. PI collects relevant plain-text data and pre-processes the data into structured sequential data. Statistical data analysis is used to understand the data and mine frequent sequential patterns. The main contribution of PI is, it demonstrates that a combination of NLP techniques and frequent sequential pattern mining can be used for finding patterns in a specific domain from unstructured natural-language text, i.e., news articles. With a minimum manual selection effort, we use these patterns to generate domain-specific scenario-based document summaries. We have applied the method in two domains. The evaluation results show that scenario-based summarization can serve to filter out irrelevant documents and also extract important sentences from relevant documents as summaries for pre-defined scenarios in a specific domain. For document level filtering, this method achieves very high precision while keeping quite high recall in both domains in our study. This demonstrates that this method may solve the problems for scenario-based document filtering in a specific domain.

Chapter 3 introduces PII and PIII to answer RQ2. PII and PIII present experiments with supervised learning for labelling business-news documents with multiple industry sectors. In much research on supervised classification it is traditional to assume not only that the test data has the same distribution of labels as the training data, but also that the classifier will be applied in the future to data drawn from the same distribution. However, this is not always the case: the label distribution may change over time, even within the same news stream. Furthermore, a single set of classifiers may be required to label data from multiple sources, such as a variety of news feeds. We are interested in exploring classification in a real-world setting, where the distribution of labels may change dynamically over time.

Therefore, one of our goals is to build classifiers that are not biased toward the particular distribution of labels in a given training set. Rather than using all available documents from a training set, we experiment with smaller subsets of balanced data. We use a balancing procedure, suitable for the multi-label setting. Using a collection of test sets, with different label distributions, we demonstrate that classifiers trained on balanced data perform better, on average, than classifiers trained using the original distribution of labels in the corpus. The main contribution of these papers is that combining a named-entity-based rote classifiers with the balanced classifiers yields better results than either classifier alone. This method improves on the best score previously reported, while using the same amount of training data for the rote classifier, and considerably less for the statistical classifiers. Experiments also shows that using company descriptors inferred from the knowledge base does not improve performance in comparison with using descriptors and company names extracted from the document.

Chapter 4 presents PIV and PV to answer RQ3. The nature of the complex relationships among traditional news, social media, and stock price fluctuations is the subject of active research. Recent studies in the area demonstrate that it is possible to find some correlation between stock prices and news, when the news are properly classified [15, 16]. We believe that the combined analysis of information can be of particular interest for specialists in various areas: business analysts, Web scientists, data journalists, etc. PIV and PV present experiments and use case studies to demonstrate interesting correlations between news and social media contents. We focus on numerical measurement and analysis of the content. Information extracted from on-line news by means of deep linguistic analysis is used to construct queries to various social media platforms. The main contribution of PIV and PV is that we combine NLP with social media analysis, and discover interesting correlations between news and social media. The results presented in the papers demonstrate the utility of collecting and comparing data from a variety of sources. In the first study, we have demonstrated that the impact that new-product events have on Twitter depends more on the industry sector than on a particular company. In the second experiment we chose three companies—Alstom, Malaysia Airlines, and General Motors to perform visual analysis of correspondence between Wikipedia views, news hits and stock prices. In the third experiment, we choose eleven big companies from different industry sectors, namely Alibaba, Alstom, Burger King, General Motors, IBM, Malaysia Airlines, Medtronic, Mt. Gox, Netflix, The Home Depot, and Xiaomi. For each of these companies we collect two time series: daily news mentions and Wikipedia views. Then we calculate

the cross-correlation between all possible pairs in this dataset to identify time-series correlations between news hits and Wikipedia views.

Based on the above research topics, Chapter 5 introduces PVI to answer RQ4. PVI presents the design and architecture of P-BDSS, which aims to extract, analyze and organize information from plain-text news and deliver it to users in a concise and easy-to-access form. The system differs from common search engines in that it uses semantic analysis, information extraction, and other methods from artificial intelligence as opposed to operating at the level of keywords for representing both news items and user queries. That is, rather than operating at the level of keywords found in a text, this system operates at a higher conceptual level, which enables richer and more informative search. This is much closer to "text understanding" than the techniques employed by traditional search engines, which are well known to be difficult to use for this type of knowledge-intensive tasks. As shown in Figure 5.1, the overall expectation is to ensure that all the modules in P-BDSS can function as described in PVI, and together form a complete decision-support system.

- As much as possible raw business data are gathered by document filtering from multiple sources.
- Raw plain-text data are processed by information extraction to produce structured information with good quality.
- Machine learning classifiers are used to decide the relevance and sector for the structured information.
- All extracted information is effectively presented by the state-of-the-art IV tools, and together with other communication and decision-support tools, provide useful decision support for BI users.

In the long run, we are driven to improve the quality of both IE and ML methods to bring P-BDSS to a higher standard. The main contribution of PVI is that it presents a novel business DSS, from a high-level perspective, gradually moving down toward the lower-level details. It demonstrates how different technologies can be combined in one system to serve the real users' information requirements in the business domain.

# References


[1] I. E. Fisher, M. R. Garnsey, and M. E. Hughes, "Natural language processing in accounting, auditing and finance: A synthesis of the literature with a roadmap for future research," *Intelligent Systems in Accounting, Finance and Management*, 2016.

[2] http://www.domo.com/, 2012.

[3] R. Gaizauskas and Y. Wilks, "Information Extraction: Beyond Document Retrieval," *Computational Linguistics and Chinese Language Processing*, vol. 3, no. 2, pp. 17–60, 1998.

[4] D. Appelt, E., J. Hobbs, R., J. Bear, D. Israel, J., and M. Tyson, "Fastus: A finite-state processor for information extraction from real-world text," *IJCAI*, pp. 1172–1178, 1993.

[5] D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks, "Named entity recognition from diverse text types," in *Recent Advances in Natural Language Processing 2001 Conference*, Bulgaria, 2001.

[6] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "Gate: A framework and graphical development environment for robust nlp tools and applications," in *In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[7] T. Jayram, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, and H. Zhu, "Avatar information extraction system," *IEEE Data Engineering Bulletin*, vol. 29, pp. 40–48, 2006.

[8] W. Shen, A. Doan, J. Naughton, F., and R. Ramakrishnan, "Declarative information extraction using datalog with embedded extraction predicates," *VLDB*, pp. 1033–1044, 2007.

[9] E. Riloff, "Automatically constructing a dictionary for information extraction tasks," in *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, 1993, pp. 811–816.

[10] E. Riloff, "Automatically generating extraction patterns from untagged text," in *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, 1996, pp. 1044–1049.

[11] S. Patwardhan and E. Riloff, "Learning domain-specific information extraction patterns from the web," in *ACL 2006 Workshop on Information Extraction Beyond the Document*, 2006.

[12] R. Sprague, H. and E. Carlson, D., *Building effective decision support systems*. Prentice-Hall, 1982.

[13] C. J. V. Rijsbergen, in *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.

[14] S. McConnell, *Rapid Development*. Microsoft Press, 1996.

[15] P. C. Tetlock, "Giving content to investor sentiment: The role of media in the stock market," *The Journal of Finance*, vol. 62, no. 3, 2007.

[16] J. Boudoukh, R. Feldman, S. Kogan, and M. Richardson, "Which news moves stock prices? A textual analysis," National Bureau of Economic Research, Tech. Rep., 2013.

[17] "Introduction to gnu wget," http://www.gnu.org/software/wget/, 2012.

[18] "Gnu grub," http://www.gnu.org/software/grub/, 2012.

[19] "Read comfortably-anytime, anywhere," http://readability.com/, 2012.

[20] Google, "Freebase data dumps," https://developers.google.com/freebase/data, 2015.

[21] Bloomberg, "Bloomberg," 2017. [Online]. Available: https://www.bloomberg.com/company/

[22] EMM, "WWW-Overview of Europe Media Monitor," 2011. [Online]. Available: http://emm.newsbrief.eu/overview.html

[23] T. Joachims, *Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21–23, 1998 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, ch. Text categorization with Support Vector Machines: Learning with many relevant features, pp. 137–142. [Online]. Available: http://dx.doi.org/10.1007/BFb0026683

[24] Z. Li, Z. Xiong, Y. Zhang, C. Liu, and K. Li, "Fast text categorization using concise semantic analysis," *Pattern Recogn. Lett.*, vol. 32, no. 3, pp. 441–448, Feb. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2010.11.001

[25] T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms*. Springer, 2002.

[26] H. Schütze, D. A. Hull, and J. O. Pedersen, "A comparison of classifiers and document representations for the routing problem," in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '95. New York, NY, USA: ACM, 1995, pp. 229–237. [Online]. Available: http://doi.acm.org/10.1145/215206.215365

[27] P. J. García-Laencina, J.-L. Sancho-Gómez, A. R. Figueiras-Vidal, and M. Verleysen, "K nearest neighbours with mutual information for simultaneous classification and missing data imputation," *Neurocomput.*, vol. 72, no. 7-9, pp. 1483–1493, Mar. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2008.11.026

[28] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *ICML*, vol. 97, 1997, pp. 412–420.

[29] C.-H. Lee and H.-C. Yang, "A classifier-based text mining approach for evaluating semantic relatedness using support vector machines," in *International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II*, vol. 1, April 2005, pp. 128–133 Vol. 1.

[30] M. Shimbo, T. Yamasaki, and Y. Matsumoto, "Automatic classification of sentences in the medline abstracts: A case study of the power of word sequence features," in *The 6th Sanken (ISIR) International Symposium*, Osaka, Japan, 2003.

[31] S. Jaillet, A. Laurent, and M. Teisseire, "Sequential patterns for text categorization," *Intell. Data Anal.*, vol. 10,

no. 3, pp. 199–214, May 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1165444.1165446

[32] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen, "Automatic acquisition of domain knowledge for information extraction," in *Proceedings of the 18th Conference on Computational Linguistics - Volume 2*, ser. COLING '00. Stroudsburg, PA, USA: Association for Computational Linguistics, 2000, pp. 940–946. [Online]. Available: http://dx.doi.org/10.3115/992730.992782

[33] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen, "Unsupervised discovery of scenario-level patterns for information extraction," in *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ser. ANLC '00. Stroudsburg, PA, USA: Association for Computational Linguistics, 2000, pp. 282–289. [Online]. Available: http://dx.doi.org/10.3115/974147.974186

[34] R. Yangarber, "Counter-training in discovery of semantic patterns," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ser. ACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 343–350. [Online]. Available: http://dx.doi.org/10.3115/1075096.1075140

[35] F. Bodon, "A fast apriori implementation," in *In Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, 2003.

[36] MathWorks, "Supervised learning workflow and algorithms," se.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html, 2016.

[37] D. Koller and M. Sahami, "Hierarchically classifying documents using very few words," Stanford InfoLab, Technical Report 1997–75, February 1997.

[38] E. D'hondt, S. Verberne, N. Weber, C. Koster, and L. Boves, "Using skipgrams and pos-based feature selection for patent classification," *Computational Linguistics in the Netherlands*, 2012.

[39] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of TF*IDF, LSI and multi-words for text classification," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2758 – 2765, 2011.

[40] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *Journal of Machine Learning Research*, vol. 3, pp. 1289–1305, Mar. 2003.

[41] D. Tikk and G. Biró, "Experiments with multi-label text classifier on the Reuters collection," in *Proceedings of the International Conference on Computational Cybernetics (ICCC 03)*, 2003, pp. 33–38.

[42] Y. Liu, H. T. Loh, and A. Sun, "Imbalanced text classification: a term weighting approach," *Expert Systems with Applications*, vol. 36, no. 1, pp. 690–701, 2009.

[43] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," in *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2007, pp. 3–24. [Online]. Available: http://dl.acm.org/citation.cfm?id=1566770.1566773

[44] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, pp. 10–18, 2009.

[45] S. Dendamrongvit, P. Vateekul, and M. Kubat, "Irrelevant attributes and imbalanced classes in multi-label text-categorization domains," *Intelligent Data Analysis*, vol. 15, no. 6, pp. 843–859, 2011.

[46] Y. Yang, "An evaluation of statistical approaches to text categorization," *Information retrieval*, vol. 1, no. 1-2, pp. 69–90, 1999.

[47] M. M. Cisse, N. Usunier, T. Arti, and P. Gallinari, "Robust Bloom filters for large multilabel classification tasks," in *Advances in Neural Information Processing Systems*, 2013, pp. 1851–1859.

[48] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *The Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[49] A. Moschitti, Q. Ju, and R. Johansson, "Modeling topic dependencies in hierarchical text categorization," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 759–767.

[50] A. Puurula, "Scalable text classification with sparse generative modeling," in *PRICAI 2012: Trends in Artificial Intelligence*, ser. Lecture Notes in Computer Science, P. Anthony, M. Ishizuka, and D. Lukose, Eds. Springer Berlin Heidelberg, 2012, vol. 7458, pp. 458–469.

[51] D. Zhuang, B. Zhang, Q. Yang, J. Yan, Z. Chen, and Y. Chen, "Efficient text classification by weighted proximal SVM," in *Fifth IEEE International Conference on Data Mining*, 2005.

[52] M. Du, J. Kangasharju, O. Karkulahti, L. Pivovarova, and R. Yangarber, "Combined analysis of news and Twitter messages," in *Joint Workshop on NLP&LOD and SWAIE: Semantic Web, Linked Open Data and Information Extraction*, 2013.

[53] W. Guo, H. Li, H. Ji, and M. T. Diab, "Linking tweets to news: A framework to enrich short text data in social media." in *Proceedings of ACL-2013*, 2013.

[54] H. Tanev, M. Ehrmann, J. Piskorski, and V. Zavarella, "Enhancing event descriptions through Twitter mining," in *Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[55] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?" in *Proceedings of the 19th international conference on World Wide Web*. ACM, 2010.

[56] N. Kanhabua, T. N. Nguyen, and C. Niederée, "What triggers human remembering of events? A large-scale analysis of catalysts for collective memory in Wikipedia," in *Joint Conference on Digital Libraries (JCDL), IEEE/ACM*, 2014.

[57] E. J. Ruiz, V. Hristidis, C. Castillo, A. Gionis, and A. Jaimes, "Correlating financial time series with micro-blogging activity," in *Proceedings of the fifth ACM international conference on Web search and data mining*, 2012.

[58] G. Wang, T. Wang, B. Wang, D. Sambasivan, Z. Zhang, H. Zheng, and B. Y. Zhao, "Crowds on Wall Street: Extracting value from collaborative investing platforms," in *Proceedings of CSCW*, 2015.

[59] E. Marsh and D. Perzanowski, "MUC-7 Evaluation Of IE Technology: Overview Of Results," *MUC*, 1998.

[60] ACE, "Automatic Content Extraction (ACE) Evaluation," 2011. [Online]. Available: http://www.itl.nist.gov/iad/mig/tests/ace/

[61] FlipDog, "WWW-Find Local Jobs & Employment Listings Job Search," 2011. [Online]. Available: www.flipdog.com/

[62] M. Ojala, "WWW-WhizBang! Labs Closes Its Doors," 2002. [Online]. Available: http://newsbreaks.infotoday.com/nbreader.asp?ArticleID=17168

[63] R. Grishman, S. Huttunen, and R. Yangarber, "Real-Time Event Extraction for Infectious Disease Outbreaks," in *In Proceedings of the 3rd Annual Human Language Technology Conference HLT-2002*, San Diego, CA, 2002.

[64] S. Patwardhan and E. Riloff, "Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions," in *EMNLP-CoNLL'07*, 2007, pp. 717–727.

[65] W. Phillips and E. Riloff, "Exploiting Role-Identifying Nouns and Expressions for Information Extraction," in *Proceedings of the 2007 Conference on Recent Advances in Natural Language Processing (RANLP-07)*, 2007.

[66] M. Krallinger and A. Valencia, "Text-mining and information-retrieval services for molecular biology," *Genome Biology*, vol. 6, no. 224, pp. 1–2,5, 2005.

[67] I. Donaldson, J. D. Martin, B. de Bruijn, C. Wolting, V. Lay, B. Tuekam, S. Zhang, B. Baskin, G. D. Bader, K. Michalickova, T. Pawson, and C. W. V. Hogue, "PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine," *BMC Bioinformatics*, pp. 4–11, 2003.

[68] M. Temkin, J. and R. Gilder, M., "Extraction of protein interaction information from unstructured text using a context-free grammar," *Bioinformatics*, vol. 19, pp. 2046–2053, 2003.

[69] D. Otasek, K. Brown, and I. Jurisica, "Confirming protein-protein interactions by text mining," in *the 6th SIAM Conference on Text Mining*, 2006.

[70] M. Atkinson, M. Du, J. Piskorski, H. Tanev, R. Yangarber, and V. Zavarella, "Techniques for multilingual security-related event extraction from online news," in *Computational Linguistics-Applications*. Springer Verlag, 2013.

[71] A. Hogenboom, F. Hogenboom, F. Frasincar, K. Schouten, and O. van der Meer, "Semantics-based information extraction for detecting economic events," *Multimedia tools and applications*, vol. 64, no. 1, 2013.

[72] T. Loughran and B. McDonald, "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks," *The Journal of Finance*, vol. 66, no. 1, 2011.

[73] A. Bodnaruk, T. Loughran, and B. McDonald, "Using 10-K text to gauge financial constraints," *J. of Financial and Quantitative Analysis*, vol. 50, no. 04, 2015.

[74] H. Sakai and S. Masuyama, "Polarity assignment to causal information extracted from financial articles concerning business performance of companies," in *Research and Development in Intelligent Systems XXV*, 2009.

[75] H. Saggion and A. Funk, "Extracting opinions and facts for business intelligence," *RNTI Journal*, vol. E17, 2009.

[76] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala, "Good debt or bad debt: Detecting semantic orientations in economic texts," *Journal of the Association for Information Science and Technology*, vol. 65, no. 4, 2014.

[77] S. Soderland, D. Fisher, J. Aseltine, and W. G. Lehnert, "Crystal: Inducing a conceptual dictionary," *CoRR*, 1995.

[78] G. Soderland, S., "Learning text analysis rules for domain-specific natural language processing," University of Massachusetts, Amherst, MA, USA, Tech. Rep., 1996.

[79] J. Kim and D. Moldovan, "Acquisition of linguistic patterns for knowledge-based information extraction," *IEEE Transactiops on Knowledge and Data Engineering*, vol. 7, no. 5, pp. 713–724, 1995.

[80] D. Bikel, M., S. Miller, R. Schwartz, and R. Weischedel, "Nymble: A high-performance learning name-finder," in *In Proceedings of ANLP-97*, 1997, pp. 194–201.

[81] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning Hidden Markov Model structure for information extraction," in *in Papers from the AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999, pp. 37–42.

[82] V. Borkar, R., K. Deshmukh, and S. Sarawagi, "Automatic text segmentation for extracting structured records," in *In Proceedings of ACM SIGMOD International Conference on Management of Data*, Santa Barabara, USA, 2001.

[83] E. Agichtein and V. Ganti, "Mining reference tables for automatic text segmentation," in *In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, USA, 2004.

[84] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman, "Exploiting diverse knowledge sources via maximum entropy in named entity recognition," in *in Sixth Workshop on Very Large Corpora New Brunswick*, New Jersey, 1998.

[85] A. Ratnaparkhi, "Learning to parse natural language with maximum entropy models," *Machine Learning*, vol. 34, 1999.

[86] A. McCallum, D. Freitag, and F. Pereira, "Maximum entropy markov models for information extraction and segmentation," in *In Proceedings of the International Conference on Machine Learning (ICML-2000)*, CA, 2000, pp. 591–598.

[87] D. Klein and C. Manning, D., "Conditional structure versus conditional estimation in NLP models," in *in Workshop on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

[88] R. Malouf, "Markov models for language-independent named entity recognition," in *In Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*, 2002.

[89] S. Huttunen, A. Vihavainen, P. Etter, von, and R. Yangarber, "Relevance prediction in information extraction using discourse and lexical features," *Nordic Conference on Computational Linguistics*, 2011.

[90] S. Huttunen, A. Vihavainen, M. Du, and R. Yangarber, "Predicting the relevance of event extraction for the end user," *Multi-source, Multilingual Information Extraction and Summarization*, pp. 163–176, 2013.

[91] "WWW-Google Maps API," 2012. [Online]. Available: http://code.google.com/apis/maps/index.html

[92] "WWW-Highcharts," 2012. [Online]. Available: http://www.highcharts.com/

[93] Department of Computer Science, University of Helsinki, "WWW-BMVis graph visualisation tool," 2012. [Online]. Available: http://www.cs.helsinki.fi/group/biomine

[94] "WWW-Semantic Interoperability of Metadata and Information in unLike Environments," 2012. [Online]. Available: http://simile.mit.edu/

[95] S. Card, K., J. Mackinlay, D., and B. Shneiderman, *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, 1999.

[96] M. Marakas, G., *Decision support systems in the twenty-first century*. Prentice Hall, 1999.

[97] J. Power, D., *Decision support systems: concepts and resources for managers*. Westport, Conn., Quorum Books, 2002.

[98] E. Turban and J. Aronson, *Decision support systems and intelligent systems*. Prentice Hall PTR Upper Saddle River, 1997.

[99] A. Gachet, *Building Model-Driven Decision Support Systems with Dicodess*. VDF, 2004.

[100] S. Haag, M. Cummings, and D. McCubbrey, J., *Management Information Systems for the Information Age*. McGraw-Hill, 2004.

[101] E. Turban, J. Aronson, E., and T.-P. Liang, *Decision Support Systems and Intelligent Systems*. Prentice Hall, 2008.

[102] Mongodb, "Mongodb 3.2 a giant leap," http://www.mongodb.com/, 2016.

[103] S. Ghemawat, H. Gobioff, and S. T. Leung, "The google file system," in *In Proceedings of the nineteenth ACM symposium on Operating systems principles, ser. SOSP'03*, 2003.

[104] "High level overview of mogilefs," http://danga.com/mogilefs/, 2012.

[105] D. Beaver, S. Kumar, H. C. Li, J. Sobel, and P. Vajgel, "Finding a needle in haystack: Facebook's photo storage," in *In Proc. 9th USENIX OSDI*, 2010.